Online Appendix to "Fast Forecasting of Unstable Data Streams for On-Demand Service Platforms"

Yu Jeffrey Hu[a], Jeroen Rombouts[b]*, and Ines Wilms[c]
[a] *Daniels School of Business, Purdue University, Indiana, The United States*
[b] *Essec Business School, France*
[c] *Department of Quantitative Economics, Maastricht University, The Netherlands*

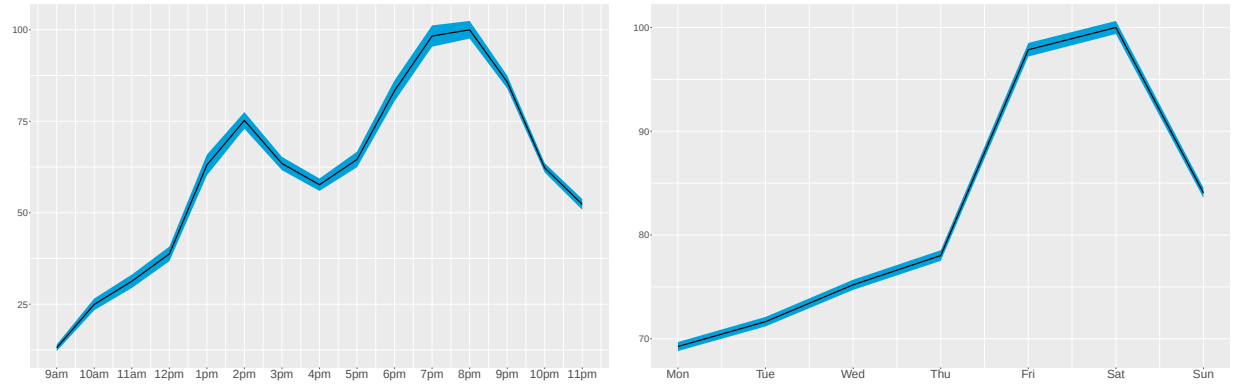
A Platform Application: Additional Data Insights

We provide additional insights into the (i) typical demand patterns of the data, and (ii) demand heterogeneity across delivery areas.

Figure A1a shows intraday UK demand. While demand is aggregated over all product categories (food, fashion, health, etc.), the large majority of deliveries consists of food. This explains why demand starts low at 9am, is moderately high around lunchtime (12-2pm), goes down only mildly between 2-5pm, sharply increases afterwards and peaks around 8pm, after which it decreases again to levels comparable to late afternoon. Apart from a strong intraday demand pattern, pronounced day of the week fluctuations are present as well, see Figure A1b. Average demand is lowest on Mondays, steadily increases until Thursdays, jumps on Fridays and Saturdays, and lowers on Sundays. These intra-day and weekly demand patterns also arise for the delivery areas albeit some areas have no reduction in late afternoon demand, or peak later than 8pm.

Figure A2 further zooms into the sparsity of the platform data, which mainly manifests itself early morning (9am to 12pm). We display histograms for average morning demand in each delivery area of the UK, Pre-Covid (top) and Post-Covid (bottom). Pre-Covid, morning demand is for

*Corresponding author. E-mail: rombouts@essec.edu, yuhu@purdue.edu, i.wilms@maastrichtuniversity.nl.



(a) Intraday Demand

(b) Day of Week Demand

Figure A1: Intraday hourly demand and day of the week average demand for UK. One standard error bands are displayed in blue.

the majority of delivery areas zero. Post-Covid, morning demand considerably increases across all delivery areas due to changing changing customer habits.

Figure A3 shows the histogram of total demand for the delivery areas over the entire sample period. An interesting tri-modal pattern emerges. The first and largest group has total demand of less than 25, the second largest group up to 50. A small third group has high demand of almost 100, the largest delivery area being Bethnal Green in the London region. Figure A4 visualizes the demand on a UK map to highlight the geographical coverage of the delivery areas. Each circle is placed at the center of one out of the 294 areas, giving by its latitude and longitude, and its diameter is proportional to the area's total demand. Unsurprisingly, most cities are covered with multiple delivery areas, the number of which is directly related to population size. Figure A5 zooms in on the London area and reveals that the delivery area total demand is not uniformly distributed. The main reason for this is that fleet planning is organized with respect to drivers' availability rather than clients' demand, thereby giving rise to substantial area-demand heterogeneity ranging from housing over business towards tourist areas.

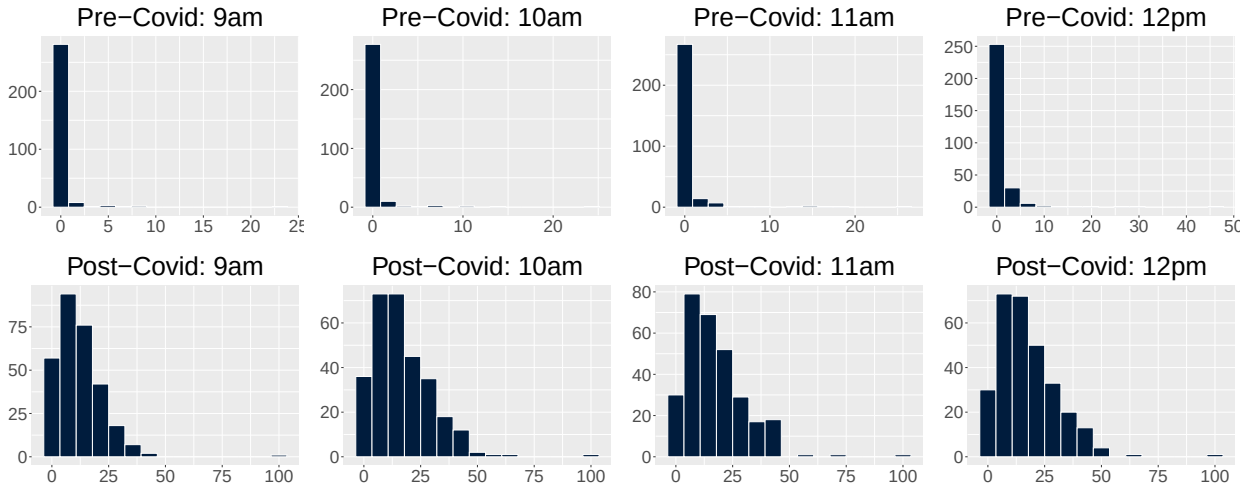


Figure A2: Histograms for average demand in morning hours in each delivery area of the UK, Pre-Covid (top) and Post-Covid (bottom).

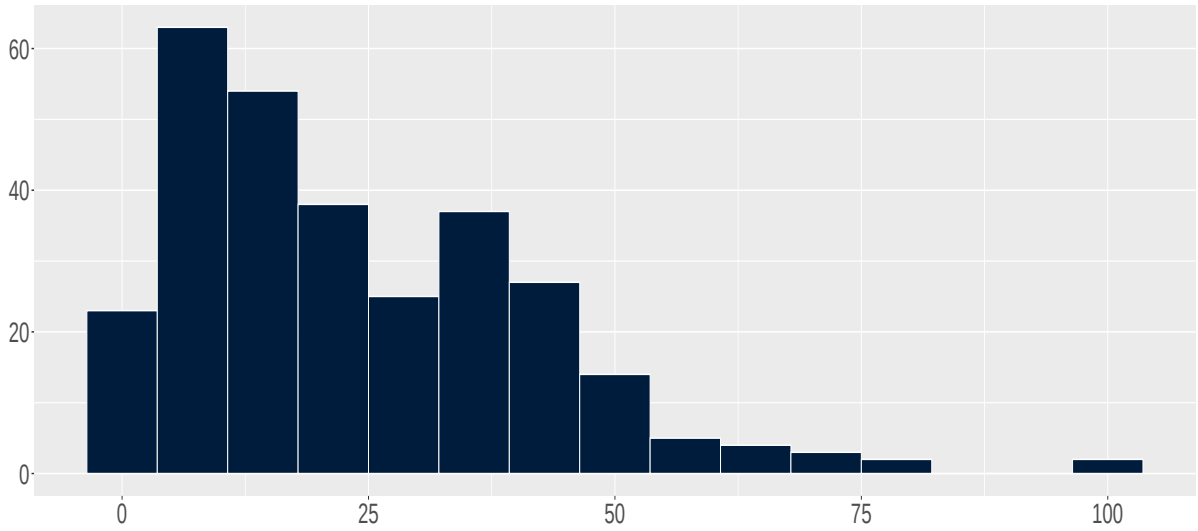


Figure A3: Histogram for aggregated demand in each delivery area of the UK.

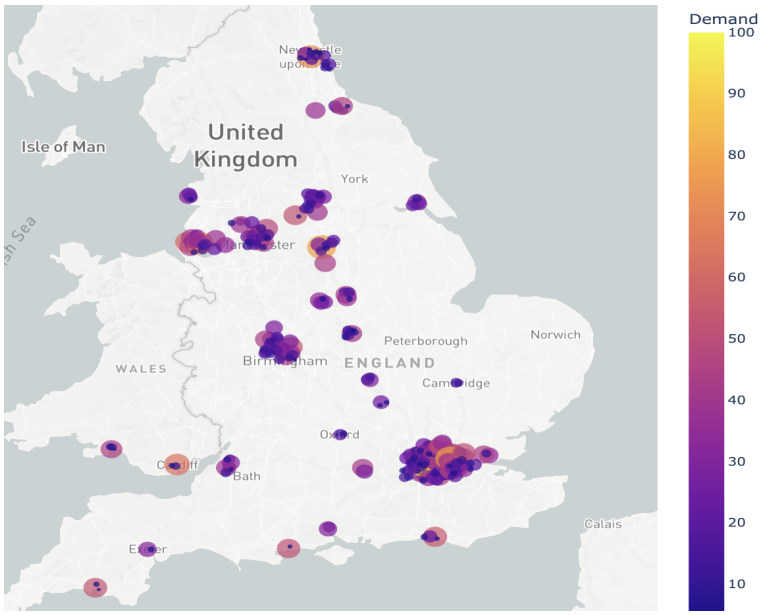


Figure A4: Aggregated demand in each delivery area in the UK.

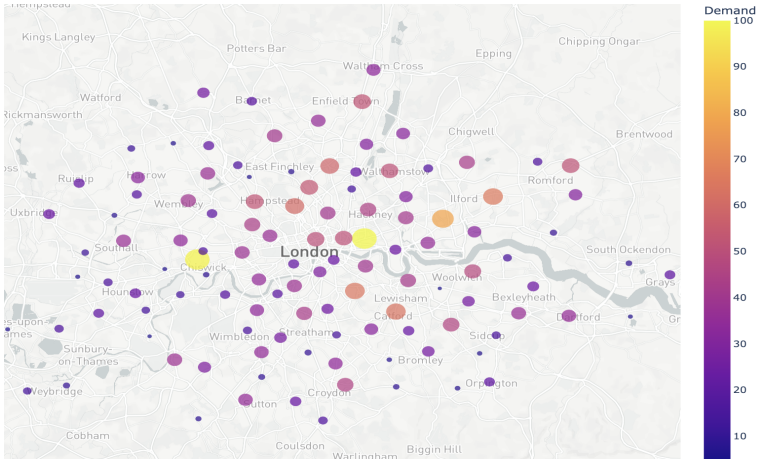


Figure A5: Aggregated demand in each delivery area in the London region.

B Benchmarks Methods

In Sections B.1 and B.2, we provide a brief description of the Prophet and LSTM forecasting approaches. We denote the demand at hourly frequency for one specific delivery area as d_t. Both approaches rely on historical information up to time t to forecast d_{t+h} where h is the forecast horizon.

B.1 Prophet

The main idea behind Prophet is decomposing d_t into the following three deterministic parts:

$$d_t = g_t + s_t + v_t + \varepsilon_t \tag{1}$$

with g_t the trend, s_t and v_t the seasonal and holiday parts respectively. The zero mean error term ε_t represents idiosyncratic changes not captured by the three parts. The advantage of this deterministic approach is that the forecast for d_{t+h} can be directly computed as $g_{t+h} + s_{t+h} + v_{t+h}$. The trend part g_t is specified by a flexible piece-wise logistic growth model accounting for change-points in the time series at known (specified by the user) or unknown dates (determined by the data only). Growth rate, capacity and offset parameters drive this trend specification. The seasonal part s_t is captured by a standard Fourier series yielding smooth effects, and for which parameter coefficients need to be estimated. Finally, holiday effects v_t are typically unsmooth and are therefore integrated in the model using indicator functions each of which are multiplied by a parameter representing the specific average effect. Assuming a law for ε_t, e.g. Gaussian, the likelihood of model (1) is known and is then combined with uninformative priors for posterior inference on the parameters. This allows integrating model parameter uncertainty and can take care of out-of-sample forecast uncertainty, but these features require additional computation time. See Taylor and Letham (2018) for more detailed explanations.

B.2 LSTM Neural Network

Long-Short-Term-Memory (LSTM) networks were first introduced by Hochreiter and Schmidhuber (1997), and have become increasingly popular for time series forecasting. They are part of recurrent neural networks but they do not suffer from the vanishing/exploding gradient problem that arises when minimising the forecast error loss function. An LSTM filters information through an input (i_t), forget (f_t), output (o_t) gate. The forget gate ensures that the input and output gates consider only the important information of the new input and previous time period respectively. A cell state c_t introduces some memory to the LSTM in order to remember the past. More formally, the structure of an LSTM to forecast d_{t+h} is a follows:

$$
\begin{aligned}
i_t &= \sigma\left(W_i X_t + U_i h_{t-1} + b_i\right) \\
f_t &= \sigma\left(W_f X_t + U_f h_{t-1} + b_f\right) \\
o_t &= \sigma\left(W_o X_t + U_o h_{t-1} + b_0\right) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh\left(W_c X_t + U_c h_{t-1} + b_c\right) \\
h_t &= o_t \odot \tanh(c_t) \\
\hat{d}_{t+h} &= W_d h_t + b_d
\end{aligned}
$$

where the weight matrices W, U, and bias vectors b contain the parameters to be estimated, X_t is the historical demand data, σ is the sigmoid function, $\odot$ is the element-wise (or Hadamard) product operator. The number of units in the LSTM is for example reflected by the dimension of the input function i_t. The initial cell state c_0 and hidden state h_0 are set to zero. Stochastic gradient descend methods are employed to minimise the squared loss $\Sigma_t(d_{t+h} - \hat{d}_{t+h})^2$ summed over the available demand data.

C Platform Application: Additional Results

Table C1: RMSE forecast performance.

	FFUDS		Naive	Prophet	LSTM		SARIMA		ETS	
	domain	data								
Breaks										
Pre-Covid	100.00	100.71	128.54	107.74	110.99	101.69	120.48	115.89	110.56	107.03
Post-Covid	100.00	101.21	119.07	138.68	116.23	100.16	111.01	106.41	106.25	102.70

Notes: One-day-ahead RMSE forecast performance of the benchmarks relative to FFUDS domain, averaged across all areas. Values above 100 indicate the percentage gain in forecast accuracy of FFUDS relative to the benchmark.

Table C2: SMAPE forecast performance for different choices of predictor sets and parameter restrictions.

Components to Vary		Variations of FFUDS					
		(1)	(2)	(3)	(4)	(5)	(6)
Predictors	Trend & seasonality						
	Lagged demand dynamics						
Restrictions	Domain-expertise						
	Data-driven						
Pre-Covid		100.00	104.75	102.06	104.96	100.95	100.11
Post-Covid		100.00	114.39	97.72	114.48	94.02	94.17

Table C3: Overview of considered change-point detection methods.

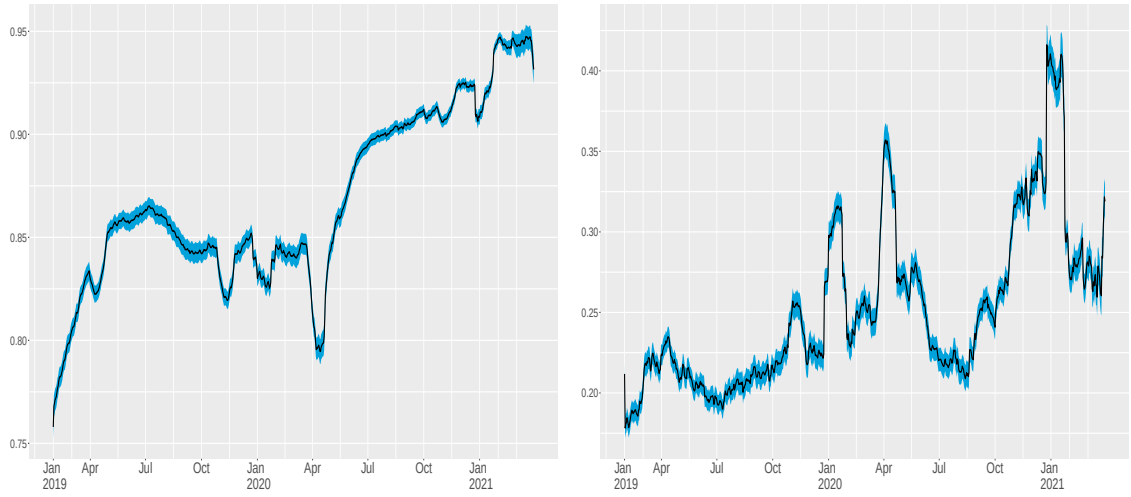
Abbreviation	Method	Reference	R-package	Package Reference
PELT	Pruned Exact Linear Time	Killick et al. (2012)	`changepoint`	Killick and Eckley (2014)
AMOC	At Most One Change	Hinkley and Hinkley (1970)	`changepoint`	Killick and Eckley (2014)
BINSEG	Binary Segmentation	Scott and Knott (1974)	`changepoint`	Killick and Eckley (2014)
BOCPD	Bayesian Online Changepoint Detection	Adams and MacKay (2007)	`ocp`	Pagotto (2019)
CPNP	Nonparametric Change Point Detection	Haynes et al. (2017)	`changepoint.np`	Haynes et al. (2022)
EPC	Energy Change Point	Matteson and James (2014)	`ecp`	James and Matteson (2014)
KCPA	Kernel Change-Point Analysis	Harchaoui et al. (2008)	`ecp`	James and Matteson (2014)
SEGNEIGH	Segment Neighborhoods	Auger and Lawrence (1989)	`changepoint`	Killick and Eckley (2014)
WBS	Wild Binary Segmentation	Fryzlewicz (2014)	`wbsts`	Korkas and Fryzlewicz (2020)

Table C4: SMAPE forecast performance for different change-point detection methods.

FFUDS with	Pre-covid	Post-covid
PELT	100.00	100.00
AMOC	99.98	99.90
BINSEG	99.96	99.81
BOCPD	100.41	99.72
CPNP	100.13	99.81
EPC	100.75	102.66
KCPA	100.75	102.66
SEGNEIGH	100.01	99.39
WBS	100.02	99.48

Table C5: SMAPE forecast performance for the change-point detection methods with multiple detected change-points, either defining the post-break sample after the earliest or the latest detected break.

FFUDS with	Pre-covid	Post-covid
PELT	100.00	100.00
SEGNEIGH, earliest break	100.01	99.39
SEGNEIGH, latest break	100.01	99.40
BOCPD, earliest break	100.41	99.72
BOCPD, latest break	100.41	99.73
CPNP, earliest break	100.13	99.81
CPNP, latest break	100.14	99.82



(a) R-squared (b) First Order AutoCorrelation Coefficient

Figure C1: Streaming average R^2 (a) and First Order AutoCorrelation Coefficient (b) for the trend and seasonality model fitted to the delivery areas. One standard error bands are displayed in blue.

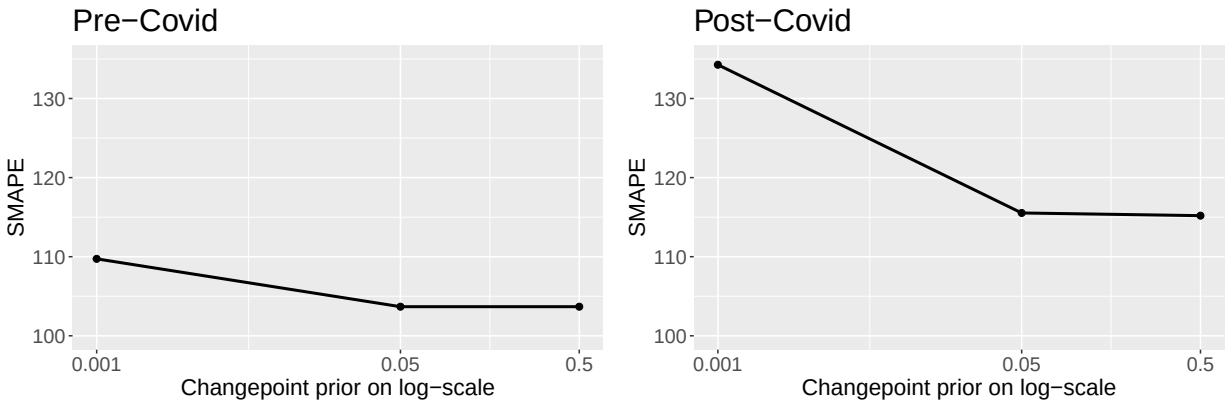


Figure C2: SMAPE forecast performance of Prophet for different change-point priors relative to FFUDS.

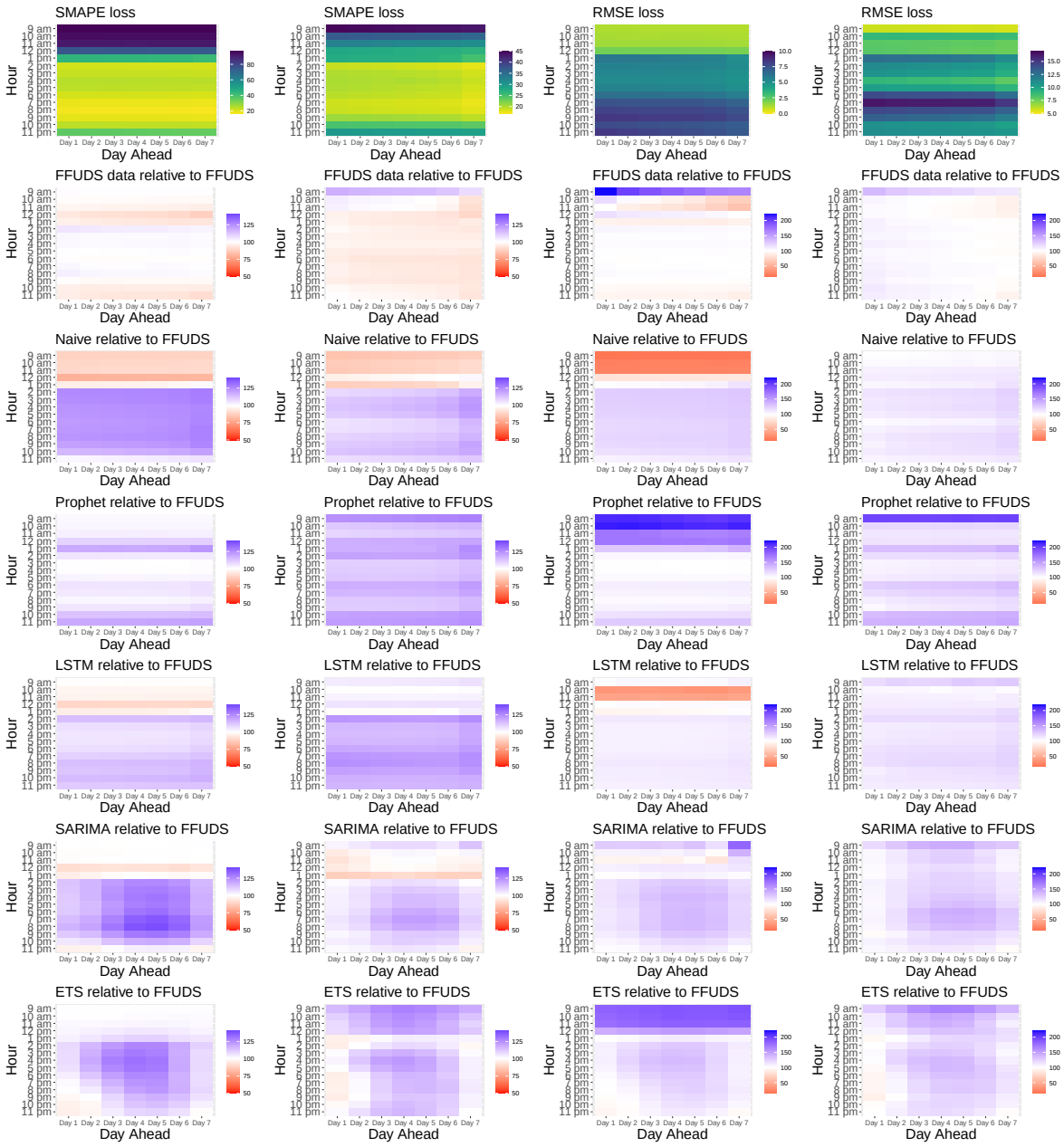


Figure C3: SMAPE and RMSE forecast performance heat maps.

Notes: Heat maps of SMAPE (columns 1-2: Pre- and Post-Covid) and RMSE (columns 3-4: Pre- and Post-Covid) forecast performance averaged across delivery areas for different hours of day (vertical axis) and days-ahead (horizontal axis). Top: Loss for FFUDS domain. Bottom rows: Respectively FFUDS data, Naive, Prophet, LSTM, SARIMA and ETS relative to FFUDS domain. Values above (below) 100% visualized in blue (red) indicate better (worse) performance of FFUDS (domain) compared to the benchmark. Equal performance is visualized in white.

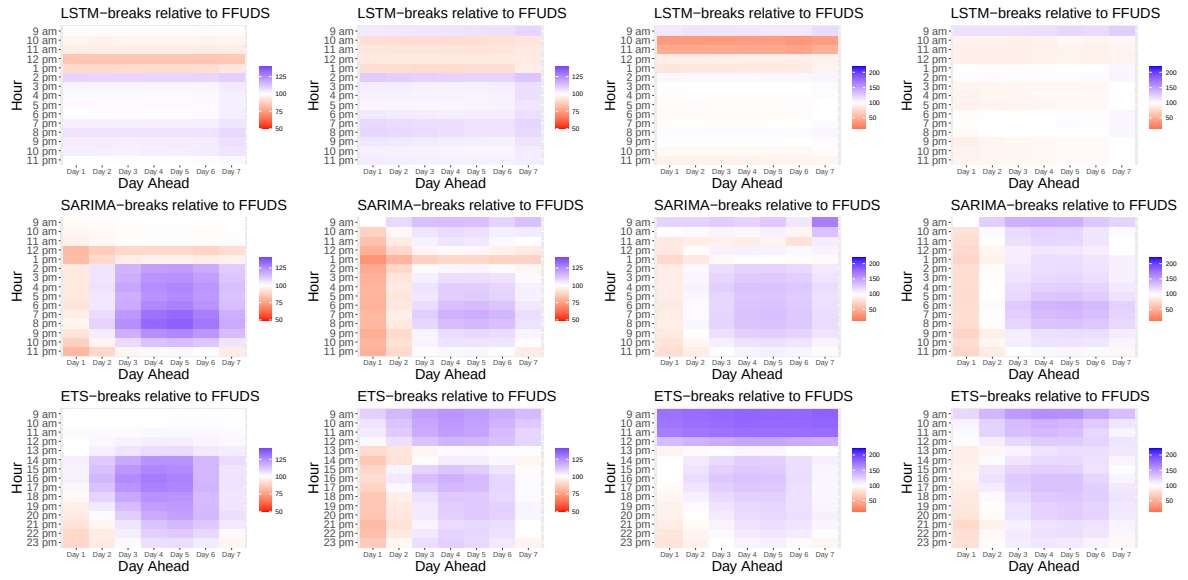


Figure C4: SMAPE and RMSE forecast performance heat maps for benchmarks with breakdown detection.

Notes: Heat maps of SMAPE (columns 1-2: Pre- and Post-Covid) and RMSE (columns 3-4: Pre- and Post-Covid) forecast performance averaged across delivery areas for different hours of day (vertical axis) and days-ahead (horizontal axis). Rows: Respectively LSTM, SARIMA and ETS all with breakdown detection relative to FFUDS (domain). Values above (below) 100% visualized in blue (red) indicate better (worse) performance of FFUDS (domain) compared to the benchmark with breakdown detection. Equal performance is visualized in white.

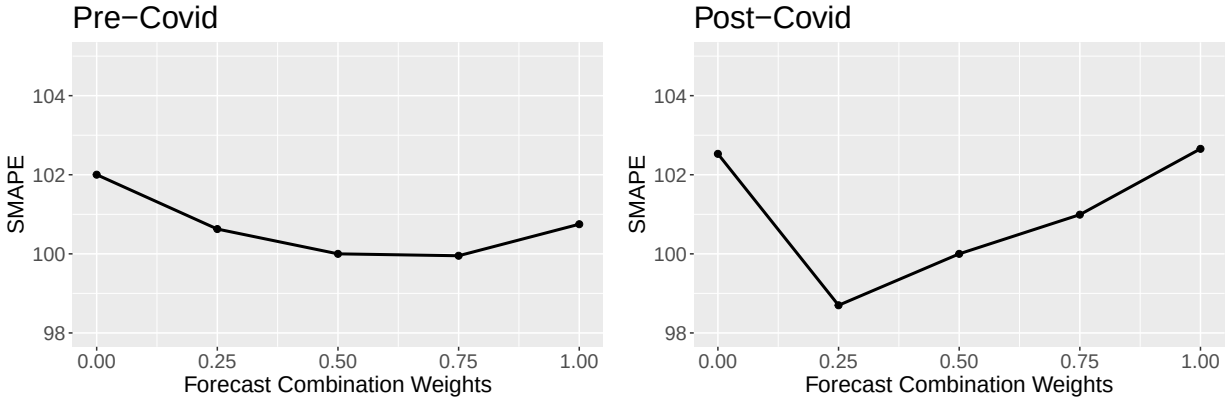


Figure C5: SMAPE forecast performance for different choices of forecast combination weight on the full-sample.

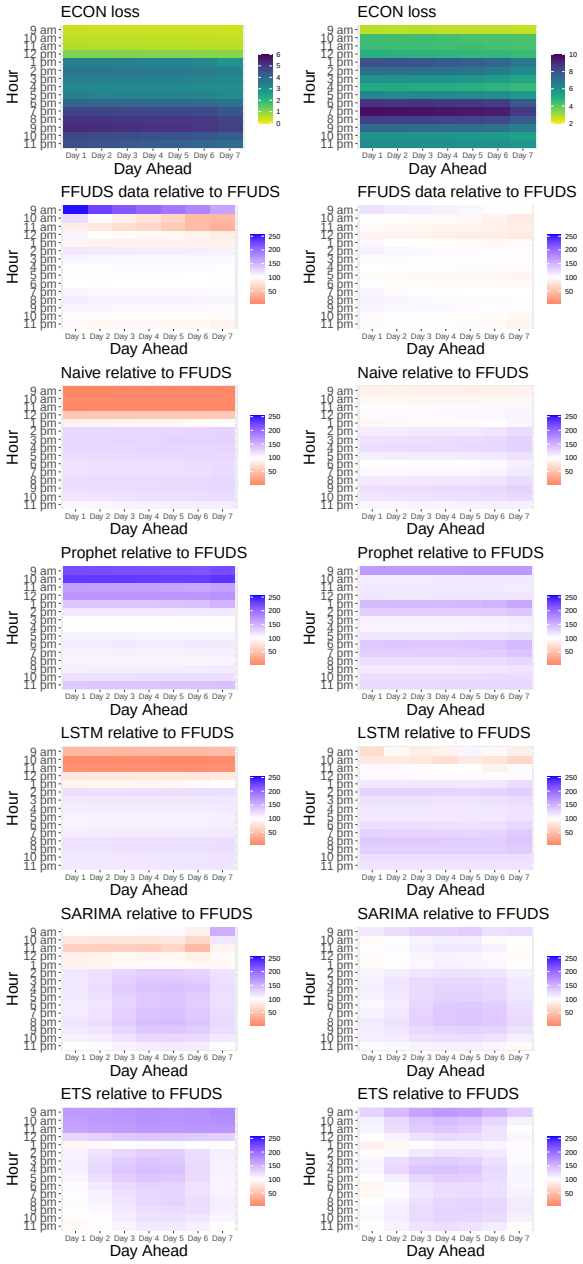


Figure C6: ECON forecast performance heat maps.

Notes: Heat maps of ECON (columns 1-2: Pre- and Post-Covid) forecast performance averaged across delivery areas for different hours of day (vertical axis) and days-ahead (horizontal axis). Top: Loss for FFUDS domain. Bottom rows: Respectively Naive, FFUDS data, Prophet, LSTM, SARIMA and ETS relative to FFUDS domain. Values above (below) 100% visualized in blue (red) indicate better (worse) performance of FFUDS (domain) compared to the benchmark. Equal performance is visualized in white.

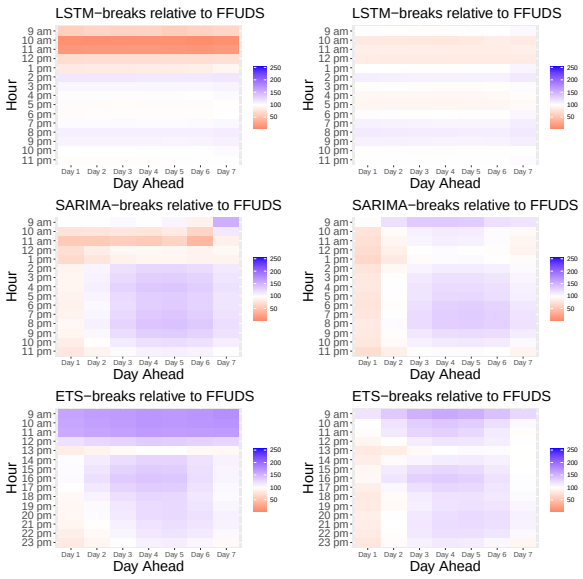


Figure C7: ECON forecast performance heat maps for benchmarks with breakdown detection.

Notes: Heat maps of ECON forecast performance averaged across delivery areas for different hours of day (vertical axis) and days-ahead (horizontal axis). Rows: Respectively LSTM, SARIMA and ETS all with breakdown detection relative to FFUDS (domain). Values above (below) 100% visualized in blue (red) indicate better (worse) performance of FFUDS (domain) compared to the benchmark with breakdown detection. Equal performance is visualized in white.

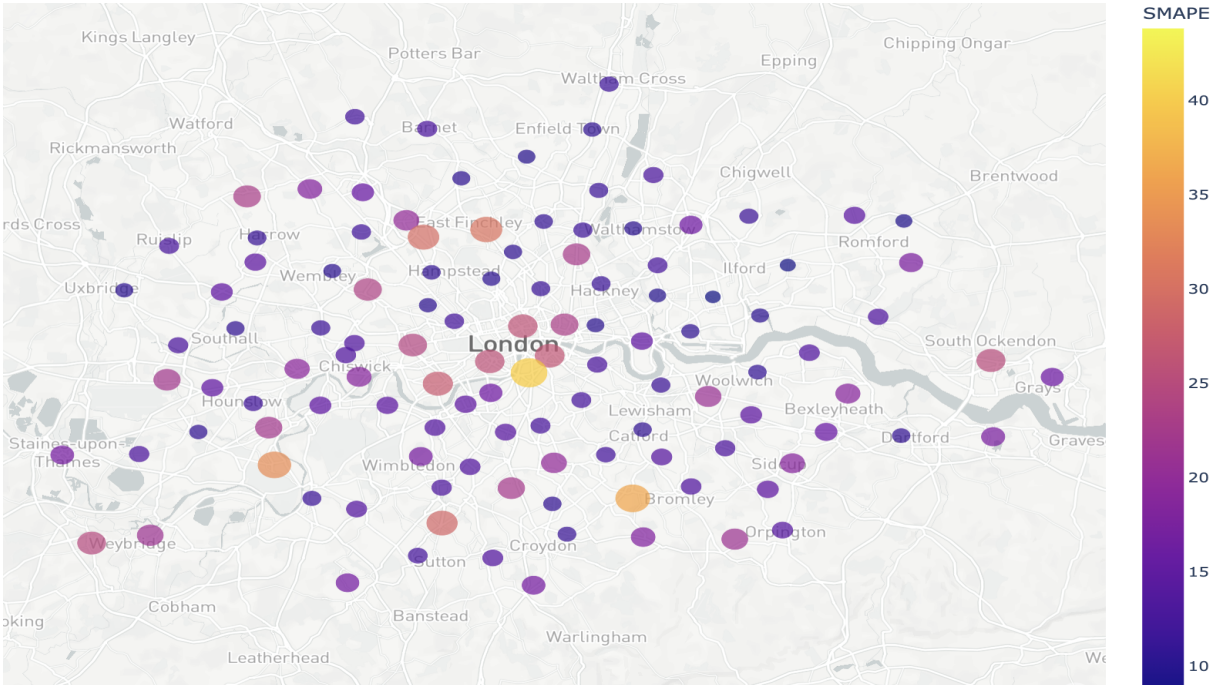


Figure C8: London region SMAPE loss; the larger the circle the larger the loss.

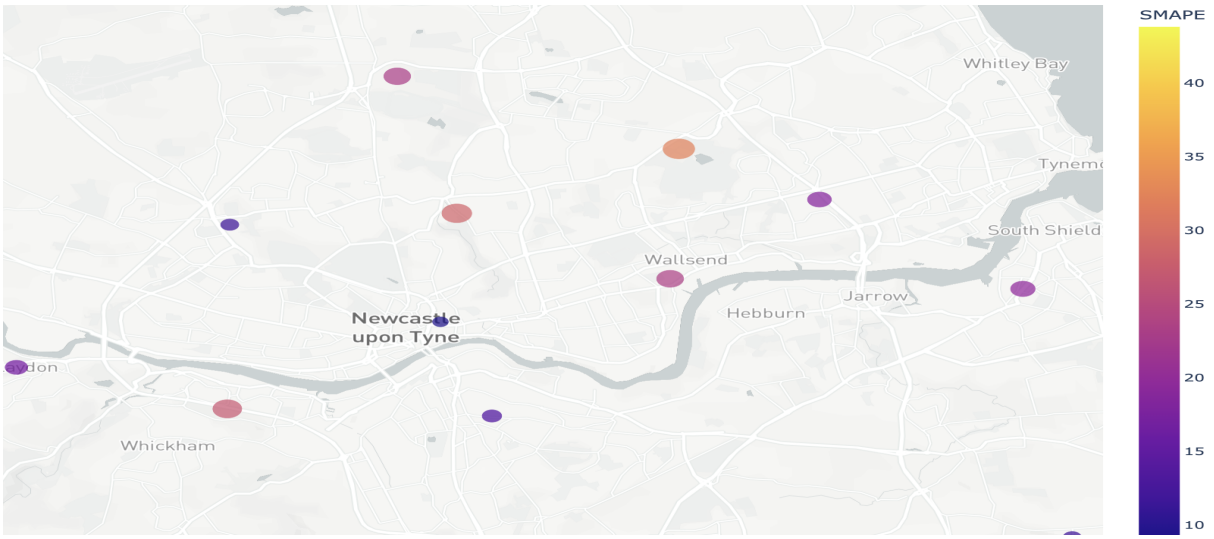


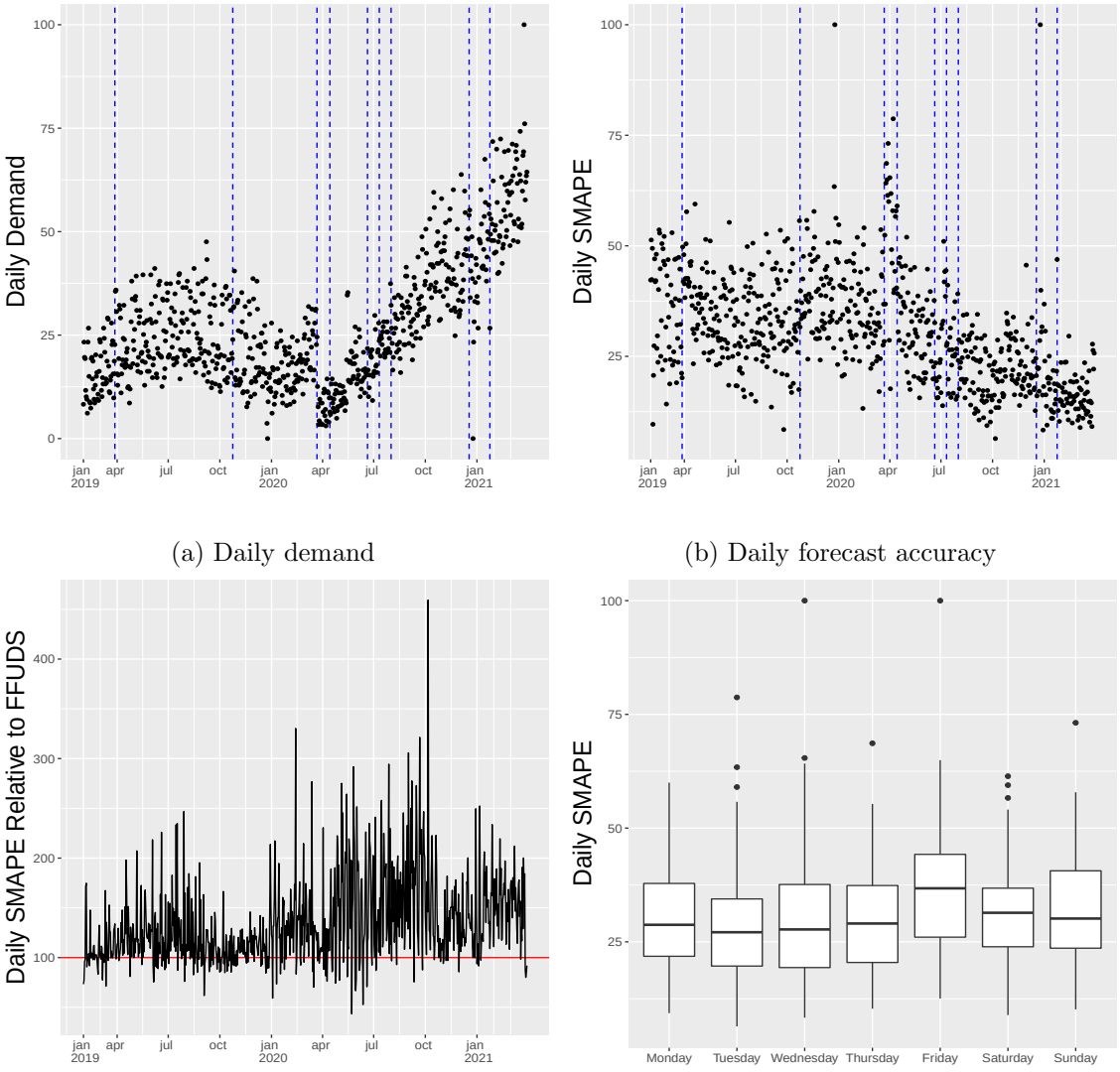
Figure C9: Newcastle region SMAPE loss; the larger the circle the larger the loss.

D Case Study of the Platform Application: Wimbledon

In this supplementary appendix, we focus on one representative area in London, namely Wimbledon, to have a clearer view on where FFUDS finds breaks and how it performs throughout the 2019-2021 sample period.

Figure D1 presents the streaming results for Wimbledon. Panel (a) shows demand aggregated to a daily frequency. We clearly see the changes in the local growth patterns, the impact of the Covid-19 first lockdown and the subsequent strong and almost linear growth. The vertical dashed blue lines indicate the nine breaks detected by FFUDS (domain) on the streaming SMAPE as displayed in panel (b). The implied break dates from SMAPE loss exhibit clear changes in forecast loss for most changes, especially towards the end of the sample where SMAPE reaches levels between 10-30% compared to 20-60% before Covid-19. Panel (c) highlights the streaming relative performance of FFUDS compared to Prophet, the platform's current forecast method, by simply dividing the respective daily losses over the sample period. In favor of FFUDS, the ratio is above 100% (red line) in extended periods, especially since July 2020. For some specific days, Prophet dominates.

The general results across forecast horizons (over all delivery areas) discussed in the main text reveal that forecast accuracy changes most throughout the business day, with small performance differences between next day and further remote weekdays. For Wimbledon, we also look at differences in forecast accuracy over the specific days of the week. Panel (d) shows boxplots of SMAPE forecast accuracy from Monday until Sunday. Except for Fridays where the median performance is slightly worse, probably due to higher volume and weekend start effects like non-regular sales, forecast performance is stable across days of the week.



(a) Daily demand

(b) Daily forecast accuracy

(c) Forecast accuracy Prophet vs. FFUDS

(d) Day of the week forecast accuracy

Figure D1: Wimbledon results.

References

Adams, R. P. and MacKay, D. J. (2007), "Bayesian online changepoint detection," *arXiv preprint arXiv:0710.3742*.

Auger, I. E. and Lawrence, C. E. (1989), "Algorithms for the optimal identification of segment neighborhoods," *Bulletin of mathematical biology*, 51, 39–54.

Fryzlewicz, P. (2014), "Wild binary segmentation for multiple change-point detection," *The Annals of Statistics*, 42, 2243–2281.

Harchaoui, Z.; Moulines, E. and Bach, F. (2008), "Kernel change-point analysis," *Advances in neural information processing systems*, 21.

Haynes, K.; Fearnhead, P. and Eckley, I. A. (2017), "A computationally efficient nonparametric approach for changepoint detection," *Statistics and computing*, 27, 1293–1305.

Haynes, K.; Killick, R.; Fearnhead, P.; Eckley, I. and Grose, D. (2022), *changepoint.np: Methods for Nonparametric Changepoint Detection*, R package version 1.0.5.

Hinkley, D. V. and Hinkley, E. A. (1970), "Inference about the change-point in a sequence of binomial variables," *Biometrika*, 57, 477–488.

Hochreiter, S. and Schmidhuber, J. (1997), "Long Short-Term Memory," *Neural Computation*, 9, 1735–1780.

James, N. A. and Matteson, D. S. (2014), "ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data," *Journal of Statistical Software*, 62, 1–25.

Killick, R. and Eckley, I. (2014), "changepoint: An R package for changepoint analysis," *Journal of Statistical Software*, 58, 1–19.

Killick, R.; Fearnhead, P. and Eckley, I. A. (2012), "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, 107, 1590–1598.

Korkas, K. and Fryzlewicz, P. (2020), *wbsts: Multiple Change-Point Detection for Nonstationary Time Series*, R package version 2.1.

Matteson, D. S. and James, N. A. (2014), "A nonparametric approach for multiple change point analysis of multivariate data," *Journal of the American Statistical Association*, 109, 334–345.

Pagotto, A. (2019), *ocp: Bayesian Online Changepoint Detection*, R package version 0.1.1.

Scott, A. J. and Knott, M. (1974), "A cluster analysis method for grouping means in the analysis of variance," *Biometrics*, 507–512.

Taylor, S. J. and Letham, B. (2018), "Forecasting at Scale," *The American Statistician*, 72, 37–45.