Delay Predictors in Multi-skill Call Centers: An Empirical Comparison with Real Data

Mamadou Thiongane

University Cheikh Anta Diop, Dakar, Senegal

Wyean Chan, Pierre L'Ecuyer

University of Montreal, Montréal QC, Canada

ICORES 2020 **Malta** February 2020

What this talk is about

Calls arrive to a call center. Different types of calls and different types of agents, with different skill sets. If a call must wait, we want to **predict** and announce its **waiting time**. Can be a point forecast (single number) or a distributional forecast (a density).

Also applies to emergency systems and various other types of service systems.

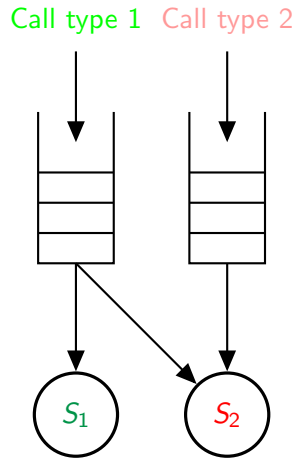
What for?

Queues in call centers are invisible. Would be useful to have information on expected wait.

Customer can make the decision to call again later instead,
or select the option to be called back by the center,
or do some other task during the wait.

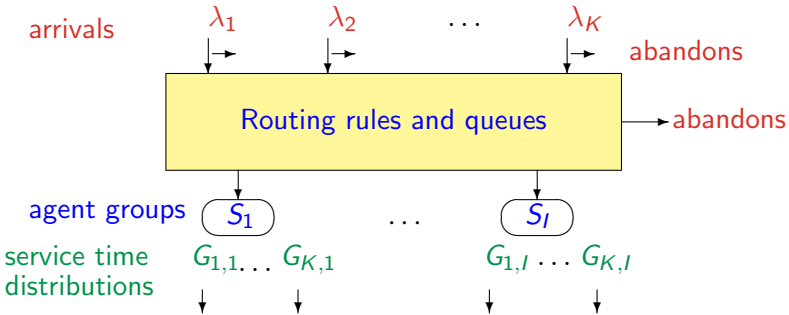
Multiskill call center

- Arriving calls have K different types, numbered from 1 to K. Each call type requires a specific skill (object of call, language, ...) and may have a different priority. In some cases, $K > 100$.
- Agents (who answer the call) are partitioned into I groups, numbered from 1 to I. Each agent in group i has skill set $S_i \subseteq \{1, 2, \ldots, K\}$.
- Routing rules: affect an arriving call to an agent if one with the right skill is available, or an agent to a waiting call when the agent becomes available.

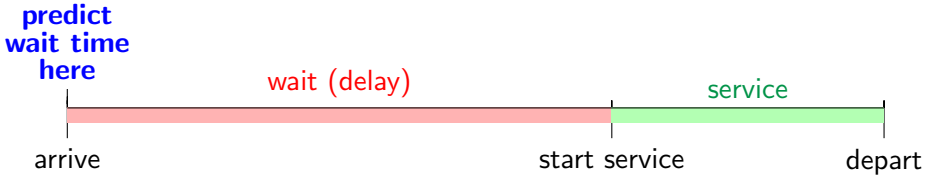


N Model

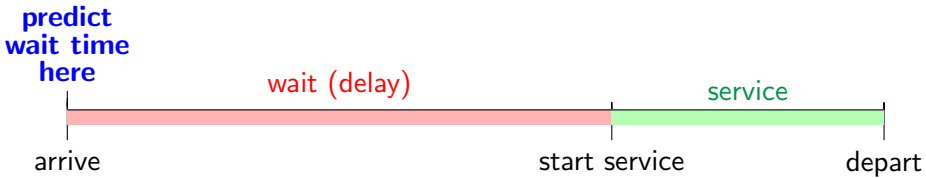
Multiskill Call Center



Service time distribution may depend on pair ⟨call type, agent group⟩.



If call is answered upon arrival, no waiting time to predict!

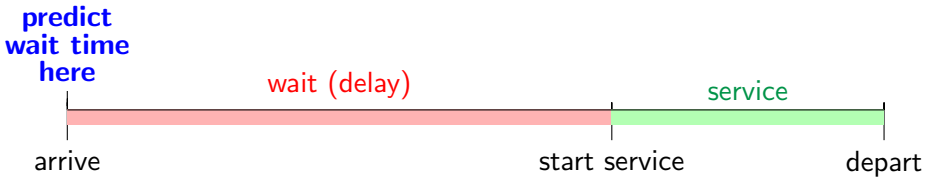


If call is answered upon arrival, no waiting time to predict!

To predict the wait time (delay), we can look at:

type of arriving call, queue lengths, waiting times of preceding calls, number of agents of relevant types, time of day, type of day, external conditions, etc.

Can make a selection among these inputs. Predictor must be computable quickly.



If call is answered upon arrival, no waiting time to predict!

To predict the wait time (delay), we can look at:

type of arriving call, queue lengths, waiting times of preceding calls, number of agents of relevant types, time of day, type of day, external conditions, etc.

Can make a selection among these inputs. Predictor must be computable quickly.

Can predict the expected waiting time conditional on current conditions.

Better but more difficult: predict conditional waiting time distribution.

How do we measure the quality of predictors

Mean squared error (MSE) of predictions for a random call of type k:

$$\mathrm{MSE}_k = \mathbb{E}[(W - P)^2]$$

where W is the realized wait time and P is the prediction.

We estimate this MSE by its empirical counterpart, the average squared error (ASE):

$$\mathrm{ASE}_k = \frac{1}{C_k} \sum_{c=1}^{C_k} (W_{k,c} - P_{k,c})^2$$

where C_k is the number of customers of type k who had to wait and got served.

How do we measure the quality of predictors

Mean squared error (MSE) of predictions for a random call of type k:

$$\mathrm{MSE}_k = \mathbb{E}[(W - P)^2]$$

where W is the realized wait time and P is the prediction.

We estimate this MSE by its empirical counterpart, the average squared error (ASE):

$$\mathrm{ASE}_k = \frac{1}{C_k} \sum_{c=1}^{C_k} (W_{k,c} - P_{k,c})^2$$

where C_k is the number of customers of type k who had to wait and got served.

We actually use a normalized version, the root relative average squared error (RRASE):

$$\mathrm{RRASE}_k = \frac{\sqrt{\mathrm{ASE}_k}}{(1/C_k) \sum_{c=1}^{C_k} W_{k,c}} \times 100.$$

Popular prediction methods

- Almost all prediction methods in the literature are for a single call type and single agent group ($K = I = 1$).
- Two main families:
  - Queue Length (QL) predictors:
    look at queue length;
  - Delay History (DH) predictors:
    look at delays of previous customers.

Examples of QL predictors

QL predictor for an M/M/s queue (s servers, service rate μ):
If C other calls are in queue when a call arrives, its expected wait is

$$P = \mathbb{E}[W \mid C] = \frac{C+1}{s\mu}.$$

Examples of QL predictors

QL predictor for an M/M/s queue (s servers, service rate μ):
If C other calls are in queue when a call arrives, its expected wait is

$$P = \mathbb{E}[W \mid C] = \frac{C+1}{s\mu}.$$

QL predictor for an M/M/s+M queue (abandonment rate ν):

$$P = \mathbb{E}[W \mid C \text{ and call get served}] = \sum_{c=1}^{C+1} \frac{1}{s\mu + c\nu}.$$

We also know the exact conditional distribution of W (gamma) in these situations.

Examples of QL predictors

QL predictor for an M/M/s queue (s servers, service rate μ):
If C other calls are in queue when a call arrives, its expected wait is

$$P = \mathbb{E}[W \mid C] = \frac{C+1}{s\mu}.$$

QL predictor for an M/M/s+M queue (abandonment rate ν):

$$P = \mathbb{E}[W \mid C \text{ and call get served}] = \sum_{c=1}^{C+1} \frac{1}{s\mu + c\nu}.$$

We also know the exact conditional distribution of W (gamma) in these situations.

Important drawback: QL predictors do not generalize to multiskill.

Examples of DH predictors

- Last-to-Enter-Service (LES): [Ibrahim and Whitt, 2009b]
  Take the delay of the last call who stated its service.

Examples of DH predictors

- Last-to-Enter-Service (LES): [Ibrahim and Whitt, 2009b]
  Take the delay of the last call who stated its service.
- Average LES (Avg-LES): [Dong et al. 2018]
  Average delay of N most recent calls who entered service after waiting.

Examples of DH predictors

- Last-to-Enter-Service (LES): [Ibrahim and Whitt, 2009b]
  Take the delay of the last call who stated its service.
- Average LES (Avg-LES): [Dong et al. 2018]
  Average delay of N most recent calls who entered service after waiting.
- Average LES per queue length (AvgC-LES): [Thiongane et al. 2016]
  Avg-LES but for N most recent calls who found the same queue length upon arrival.

Examples of DH predictors

- Last-to-Enter-Service (LES): [Ibrahim and Whitt, 2009b]
  Take the delay of the last call who stated its service.
- Average LES (Avg-LES): [Dong et al. 2018]
  Average delay of N most recent calls who entered service after waiting.
- Average LES per queue length (AvgC-LES): [Thiongane et al. 2016]
  Avg-LES but for N most recent calls who found the same queue length upon arrival.
- Extrapolated LES (E-LES): [Thiongane et al. 2016]
  Use information on elapsed waits of calls still in queue, via extrapolation.

Examples of DH predictors

- Last-to-Enter-Service (LES): [Ibrahim and Whitt, 2009b]
  Take the delay of the last call who stated its service.
- Average LES (Avg-LES): [Dong et al. 2018]
  Average delay of N most recent calls who entered service after waiting.
- Average LES per queue length (AvgC-LES): [Thiongane et al. 2016]
  Avg-LES but for N most recent calls who found the same queue length upon arrival.
- Extrapolated LES (E-LES): [Thiongane et al. 2016]
  Use information on elapsed waits of calls still in queue, via extrapolation.
- Proportional Queue LES (P-LES): [Ibrahim et al., 2016)]
  Take LES, but readjusts by the ratio of current number in system
  divided by number in system when the LES arrived.

Examples of DH predictors

- Last-to-Enter-Service (LES): [Ibrahim and Whitt, 2009b]
  Take the delay of the last call who stated its service.
- Average LES (Avg-LES): [Dong et al. 2018]
  Average delay of N most recent calls who entered service after waiting.
- Average LES per queue length (AvgC-LES): [Thiongane et al. 2016]
  Avg-LES but for N most recent calls who found the same queue length upon arrival.
- Extrapolated LES (E-LES): [Thiongane et al. 2016]
  Use information on elapsed waits of calls still in queue, via extrapolation.
- Proportional Queue LES (P-LES): [Ibrahim et al., 2016)]
  Take LES, but readjusts by the ratio of current number in system
  divided by number in system when the LES arrived.
- Head of Line (HOL):
  Delay of customer who is now at the head of the queue.
- Most Recent Completed Service (RCS):
  Delay of last call to have completed its service.

Comparing QL and DH predictors

QL predictors win for simple systems, but do not apply to multiskill or complicated systems.

DH predictors can be adapted to multiskill: One can simply consider the only the delays of previous calls of the same type.

But DH predictors perform poorly when the conditions (arrival rates, number of servers, etc.) vary significantly in time. This commonly occurs.

Predictors based on machine learning (ML)

Let $\mathbf{x}$ denote the current condition of the system, i.e., a vector of selected input variables such as queue lengths, numbers of servers having proper skill, current time, etc.

The goal is to construct a predictor function $P_{k,\theta}(\mathbf{x})$ for call type k.

This function has a vector of parameters $\boldsymbol{\theta}$ that must be learned (or estimated) from previous data, in a training step.

Predictors based on machine learning (ML)

Let $\mathbf{x}$ denote the current condition of the system, i.e., a vector of selected input variables such as queue lengths, numbers of servers having proper skill, current time, etc.

The goal is to construct a predictor function $P_{k,\theta}(\mathbf{x})$ for call type k.

This function has a vector of parameters $\boldsymbol{\theta}$ that must be learned (or estimated) from previous data, in a training step. We consider three ways of defining $P_{k,\theta}(\mathbf{x})$:

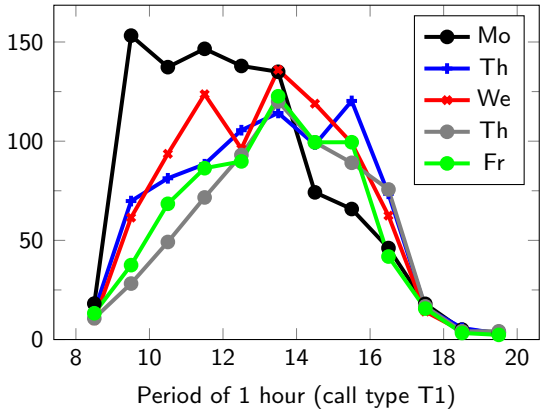
1. Cubic smoothing spline, additive in the input variables (RS);
   Uses least-squares regression with penalty term on the variation of the function.
2. Lasso linear regression (LR);
   Linear regression with penalty term equal to sum of absolute values of coefficients.
3. Feed-forward multilayer artificial neural network (ANN).
   Four or five layers, rectifier activation function $h(\mathbf{z}) = max(0, b + \mathbf{w} \cdot \mathbf{z})$ at each node. Here $\boldsymbol{\theta}$ is the set of all pairs $(b, \mathbf{w})$ in the network. Training used back-propagation algorithm with stochastic gradient descent, with Pylearn2 software.

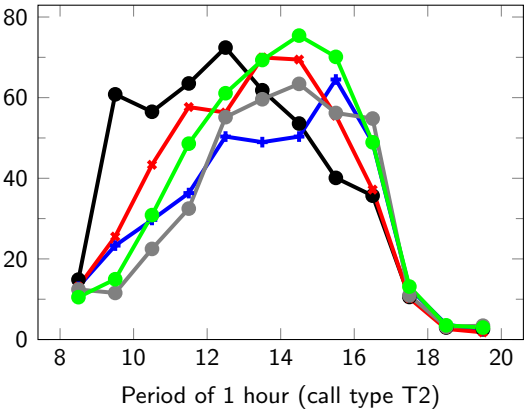
Call Center Data

We obtained one year of data (2014) from VANAD Lab call center located in Rotterdam. Operates from 8:00 to 20:00, Monday to Friday, with 27 call types, and 312 agents. There were about 1.54 million calls, of which 56% did not wait, 6% abandoned, and 38% waited and were answered (our data). We report on the 5 most frequent call types, T1 to T5.

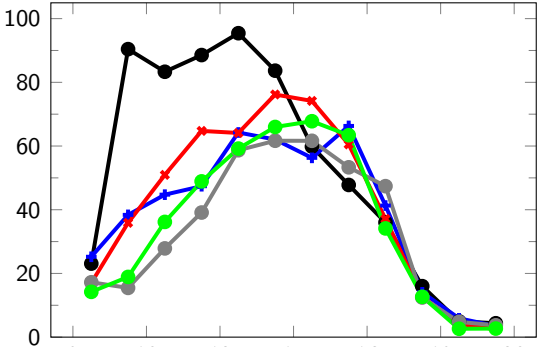
	T1	T2	T3	T4	T5
Number	568 554	270 675	311 523	112 711	25 839
No wait	61%	52%	55%	45%	34%
Wait	35%	40%	40%	46%	54%
Abandon	4%	7%	5%	8%	12%
Av. wait (sec)	77	91	83	85	110
Av. service (sec)	350	308	281	411	311
Av queue size	8.2	3.3	4.4	4.3	0.9

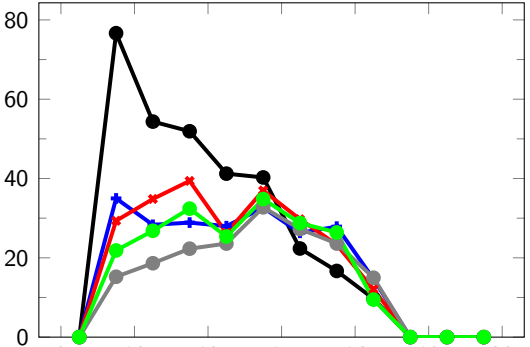
Arrival counts per hour, per type, for each type of day



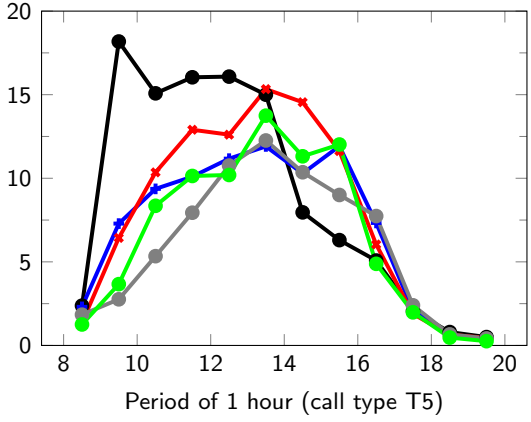


Arrival counts per hour, per type, for each type of day





Arrival counts per hour, per type, for each type of day



Monday is obviously different than other days, so we take Monday apart and regroup the other four days.

We now look at the results for Tuesday–Friday together.

How to select x?

Candidate input variables in our numerical study:

- vector of queue lengths per call type;
- vector of number of servers that can handle call type k, for each k;
- total number of agents;
- current time of day;
- wait time of the N most recently served customers of the given call type;
- wait times predicted by LES, P-LES, E-LES, Avg-LES, and AvgC-LES.

The raw data did not contain all this information (e.g., queue lengths, number of servers,...). We had to write and run a simulation to recover this info from the data we had.

How to select x?

Candidate input variables in our numerical study:

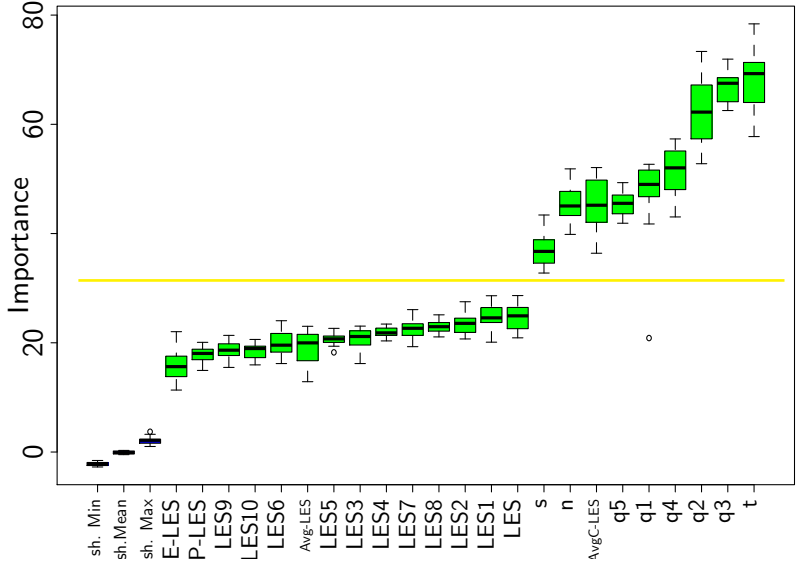
- vector of queue lengths per call type;
- vector of number of servers that can handle call type k, for each k;
- total number of agents;
- current time of day;
- wait time of the N most recently served customers of the given call type;
- wait times predicted by LES, P-LES, E-LES, Avg-LES, and AvgC-LES.

The raw data did not contain all this information (e.g., queue lengths, number of servers,...). We had to write and run a simulation to recover this info from the data we had.

We use a feature-selection method which provides importance scores for all variables in terms of its predictive power. The variables with high-enough scores are selected for the model.

The method we used is Boruta, a state-of-the-art method based on the random forest algorithm of Breiman (2001). See the paper for details.

Boruta results: important input variables for call type T1



RRASE for delay predictors with testing data

We used 80% of data to estimate the parameters, 20% to test prediction accuracy.

	Call Types				
Predictors	T1	T2	T3	T4	T5
Avg-LES	48.9	61.0	56.7	48.7	69.7
LES	44.3	57.7	51.8	44.5	66.1
AvgC-LES	44.3	56.5	51.6	42.4	62.4
E-LES	63.7	65.4	64.0	58.8	77.5
P-LES	71.2	70.5	71.4	68.5	80.3
RS	39.6	49.2	45.5	39.5	50.1
RL	41.5	51.5	47.1	38.5	51.7
ANN	36.1	46.2	44.8	37.7	48.7

Whatever method we use, there will always remain prediction error due to intrinsic randomness in the system.

What if we remove some input deemed important by Boruta

In Thiongane et al. (2015, 2016) we made some preliminary experiments with RE and ANN with data generated by a simulation program (not from a real call center). We did not use Boruta, but just fixed the inputs.

In the input variables, we did not consider for example the current time t, the AvgC-LES prediction, and the queue lengths for other call types.

What if we remove some of these inputs from previous experiment?

What if we remove some input deemed important by Boruta

In Thiongane et al. (2015, 2016) we made some preliminary experiments with RE and ANN with data generated by a simulation program (not from a real call center). We did not use Boruta, but just fixed the inputs.

In the input variables, we did not consider for example the current time t, the AvgC-LES prediction, and the queue lengths for other call types.

What if we remove some of these inputs from previous experiment?

We tried:

ANN-2: remove t and AvgC-LES.

ANN-3: remove also a queue length for another call type.

RRASE of delay predictors with important input removed

RS-2, LR-2, ANN-2: remove t and AvgC-LES.
RS-3, LR-3, ANN-3: remove also a queue length for another call type.

	Call Types				
Predictors	T1	T2	T3	T4	T5
RS	39.6	49.2	45.5	39.5	50.1
LR	41.5	51.5	47.1	38.5	51.7
ANN	36.1	46.2	44.8	37.7	48.7
RS-2	41.9	52.0	47.7	40.9	52.5
LR-2	43.9	54.0	49.1	39.2	53.1
ANN-2	39.7	49.2	46.9	38.5	50.3
RS-3	42.5	53.0	47.9	41.2	52.9
LR-3	44.3	55.4	50.7	39.8	54.0
ANN-3	40.4	50.2	47.0	38.7	50.9

Conclusion and other current work

- Deep neural networks are really effective to predict the waiting time when a customer arrives, based on well-selected input variables that describe the state of the system. On the other hand, they require a lot of data for training.

Conclusion and other current work

- Deep neural networks are really effective to predict the waiting time when a customer arrives, based on well-selected input variables that describe the state of the system. On the other hand, they require a lot of data for training.
- Boruta is quite effective in identifying important inputs.

Conclusion and other current work

- Deep neural networks are really effective to predict the waiting time when a customer arrives, based on well-selected input variables that describe the state of the system. On the other hand, they require a lot of data for training.
- Boruta is quite effective in identifying important inputs.
- It would be interesting to experiment with this methodology for a larger variety of call centers and other types of service systems, e.g., with much longer waits. This requires appropriate data!

Conclusion and other current work

- Deep neural networks are really effective to predict the waiting time when a customer arrives, based on well-selected input variables that describe the state of the system. On the other hand, they require a lot of data for training.
- Boruta is quite effective in identifying important inputs.
- It would be interesting to experiment with this methodology for a larger variety of call centers and other types of service systems, e.g., with much longer waits. This requires appropriate data!
- In addition to predicting the expected waiting time, we also want to predict the waiting time distribution (e.g., conditional density).

Some related references

- Dong, J., Yom Tov, E., and Yom Tov, G. (2018). The impact of delay announcements on hospital network coordination and waiting times. *Management Science*, 65(5):1949–2443.
- Friedman, J., Hastie, T., Tibshirani, R., Narasimhan, B., Simon, N., and Qian, J. (2019). *R Package glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models*. `https://CRAN.R-project.org/package=glmnet`.
- Goodfellow, I., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., Bergstra, J., Bastien, F., and Bengio, Y. (2013). Pylearn2: A machine learning research library.
- Ibrahim, R., L'Ecuyer, P., Shen, H., and Thiongane, M. (2016). Inter-dependent, heterogeneous, and time-varying service-time distributions in call centers. *European J. of Oper. Research*, 250:480–492.
- Ibrahim, R. and Whitt, W. (2009a). Real-time delay estimation based on delay history. *Manufacturing and Services Operations Management*, 11:397–415.
- Kursa, M. B. and Rudnicki, W. R. (2010). Feature selection with the boruta package. *Journal of Statistical Software*, 36:1–13.
- Thiongane, M., Chan, W., and L'Ecuyer, P. (2015). Waiting time predictors for multiskill call centers. In *Proc. of the 2015 Winter Simulation Conf.*, 3073–3084. IEEE Press.
- Thiongane, M., Chan, W., and L'Ecuyer, P. (2016). New history-based delay predictors for service systems. In *Proc. of the 2016 Winter Simulation Conf.*, 425–436. IEEE Press.
- Wood, S. N. (2019). *R Package mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation*. `https://CRAN.R-project.org/package=mgcv`.