Political Science Research and Methods (XXX), **XXX**, 1–33
doi:

RESEARCH NOTE

The Role of Hyperparameters in Machine Learning Models and How to Tune Them

Christian Arnold,[1] Luka Biedebach,[2] Andreas Küpfer,[3] and Marcel Neunhoeffer[4]

[1]Cardiff University
[2]Reykjavik University
[3]Technical University of Darmstadt
[4]Boston University & LMU Munich

(Received ; revised ; accepted ; first published online)

Abstract
Hyperparameters critically influence how well machine learning models perform on unseen, out-of-sample data. Systematically comparing the performance of different hyperparameter settings will often go a long way in building confidence about a model's performance. However, analyzing 64 machine learning related manuscripts published in three leading political science journals (APSR, PA, and PSRM) between 2016 and 2021, we find that only 13 publications (20.31%) report the hyperparameters and also how they tuned them in either the paper or the appendix. We illustrate the dangers of cursory attention to model and tuning transparency in comparing machine learning models' capability to predict electoral violence from tweets. The tuning of hyperparameters and their documentation should become a standard component of robustness checks for machine learning models.

Keywords: machine learning; hyperparameter optimization; good advice

1. Why Care about Hyperparameters?

When political scientists work with machine learning models, they want to find a model that generalizes well from training data to new, unseen data.[1] Hyperparameters play a key role in

1. A machine learning algorithm is "a computer program [that is] said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

this endeavor because they determine the models' capacity to generalize. Finding a good set of hyperparameters critically affects conclusions about a model's performance. The failure to correctly tune and report hyperparameters has recently been identified as a key impediment to the accumulation of knowledge in computer science (e.g. Bouthillier, Laurent, and Vincent 2019; Bouthillier et al. 2021; Cooper et al. 2021; Henderson et al. 2018; Gundersen, Coakley, and Kirkpatrick 2022; Melis, Dyer, and Blunsom 2018). Is political science making the same mistake?

We examined 64 machine learning-related papers published between 1 January 2016 and 20 October 2021 in some of the top journals of our discipline—the American Political Science Review (APSR), Political Analysis (PA), and Political Science Research and Methods (PSRM). Of the 64 publications we analyzed, 36 (56.25%) do not report the values of their hyperparameters, neither in the paper nor the appendix. Forty-nine publications (76.56%) do not share information about how they used tuning to find the values of their hyperparameters. Only 13 publications (20.31%) offer a complete account of the hyperparameters and their tuning. Not being transparent is a dangerous habit because readers and reviewers cannot assess the quality of a manuscript without access to the replication code.

With this paper, therefore, we raise the awareness that hyperparameters and their tuning matter. In statistical inference, the goal is to estimate the value of an unknowable population parameter. Including robustness checks in a paper and its appendix is good practice, allowing others to understand critical choices in research design and statistical modeling. The actual out-of-sample performance of a machine learning model is such an unknown quantity, too. We suggest handling estimates of population parameters and hyperparameters in machine learning models with the same loving care.

First, we explain what hyperparameters are and why they are essential. Second, we show why it is dangerous not to be transparent about hyperparameters. Third, we offer best practice advice about properly selecting hyperparameters. Finally, we illustrate our points by comparing the performance of several machine learning models to predict electoral violence from tweets (Muchlinski et al. 2021).

2. What Are Hyperparameters and Why Do They Need to Be Tuned?

Many machine learning models have parameters and also hyperparameters. Model parameters are learned during training, and hyperparameters are typically set before training. Hyperparameters

(Mitchell 1997)

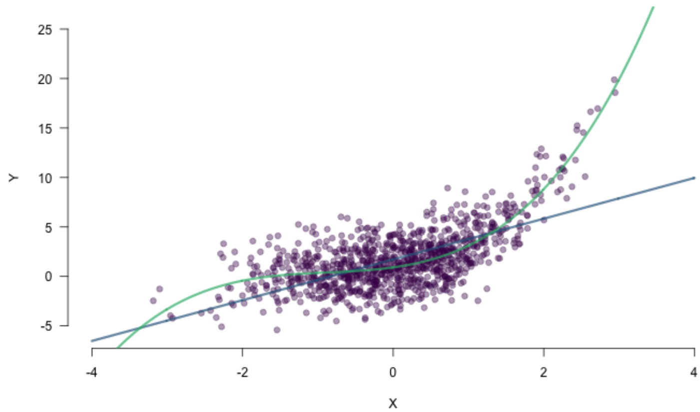

Figure 1. Example with polynomial regression. Data $X \sim N(0, 1)$. Data generating process: $Y = 1 + X + 0.8X^2 + 0.3X^3 + \epsilon$, with $\epsilon \sim N(0, 2)$. Regression Line for Bivariate OLS Model in Blue. Regression Curve for Polynomial Regression with $\lambda = 3$ in Teal.

determine how and what a model can learn and how well the model will perform on out-of-sample data. Hyperparameters are thus situated at a meta-level above the models themselves.

Consider the following stylized example displayed in Figure 1.[2] A linear regression approach could model the relationship between X and Y as $\hat{Y} = \beta_0 + \beta_1 X$. A more flexible model would include additional polynomials in X. For example, choosing $\lambda = 2$ encodes the theoretical belief that Y is best predicted by a quadratic function of X, i.e., $\hat{Y} = \beta_0 + \beta_1 X + \beta_2 X^2$. But it is also possible to rely on data only to find the optimal value of λ. Measuring the generalization error with a metric like the mean squared error helps empirically select the most promising value of λ.

This polynomial regression comes with both parameters and hyperparameters. *Parameters* are variables that belong to the model itself, in our example, the regression equation coefficients. *Hyperparameters* are those variables that help specify the exact model. In the context of the polynomial regression, λ is the hyperparameter that determines how many parameters will be learned (Goodfellow, Bengio, and Courville 2016). Machine learning models can, of course, come with many more hyperparameters that relate not only to the exact parameterization of the machine learning model. Anything part of the function that maps the data to a performance measure and that can be set to different values can be considered a hyperparameter, e.g., the choice and settings of a kernel in a support vector machine (SVM), the number of trees in a random forest (RF), or the choice of a

2. See also Shalev-Shwartz and Ben-David (2014) and Goodfellow, Bengio, and Courville (2016).

particular optimization algorithm.

3. Misselecting Hyperparameters

Research on machine learning has recently identified several problems that may arise from handling hyperparameters without care. The failure to report the chosen hyperparameters impedes scientific progress (Bouthillier, Laurent, and Vincent 2019; Bouthillier et al. 2021; Gundersen, Coakley, and Kirkpatrick 2022; Henderson et al. 2018). In the face of a hyperparameter space marked by the curse of dimensionality, other researchers can only replicate published work if they know the hyperparameters used in the original study (Sculley et al. 2018). In addition, it is essential to tune the hyperparameters of all models, including baseline models. Without such tuning, it is impossible to compare the performance of two different models M_a and M_b: While some may find that the performance of M_a is better than M_b, others replicating the study with different hyperparameter settings could conclude the opposite: that indeed M_a is *not* better than that of M_b. Such "hyperparameter deception" (Cooper et al. 2021) has confused scientific progress in various subfields in computer science where machine learning plays a key role, including natural language processing (Melis, Dyer, and Blunsom 2018), computer vision (Musgrave, Belongie, and Lim 2020), and generative models (Lucic et al. 2018). Reviewers and readers need to comprehend the hyperparameter tuning to assess whether a new model reliably performs better or whether a study tests new hyperparameters (Cooper et al. 2021).

It is good to see political scientists also discuss and stress the relevance of hyperparameter tuning in their work (e.g., Cranmer and Desmarais 2017; Fariss and Jones 2018; Miller, Linder, and Mebane 2020; Rheault and Cochrane 2020; Chang and Masterson 2020; Torres and Cantú 2021). But does the broader political science community fulfill the requirements suggested in the computer science literature? To understand how hyperparameters are used in the discipline, we searched for the term "machine learning" in all papers published in APSR, PA, and PSRM after 1 January 2016 and before 20 October 2021. Suppose a paper applies a machine learning model with tunable hyperparameters. In that case, we first annotate whether the authors report the final values of hyperparameters for all models in their paper or its appendix.[3] We also record whether authors transparently describe how they tuned hyperparameters.[4] Table 1 summarizes the findings from our annotations. We find

3. We call this "model transparency", i.e., could a reader understand the final models without access to the replication code?
4. We call this "tuning transparency", i.e., could a reader understand the hyperparameter tuning without access to the replication code? Please see Appendix 1 for more details about our annotations.

that 34 (53.12%) publications neither report the values of the final hyperparameters nor the tuning regime in the publication or its appendix. Another 15 publications (23.44%) offer information about the final hyperparameter values but not how they tuned the machine learning models. In two cases (3.12%), we find no information about the final values of the hyperparameters but about the tuning regime. Finally, only 13 publications (20.31%) offer a full account of both the final choice of the hyperparameters and the way the tuning occurred in either the paper itself or its appendix.

Note that we annotated the literature in a way that helps understand whether reviewers and readers can assess the robustness of the analyses based on the manuscript and its appendix. Our analysis does not consider the replication code since it typically does not find consideration in the review process. In addition, we do not make any judgments about correctness. A paper without information about hyperparameter values or their tuning can still be correct. Similarly, a paper that reports hyperparameter values and a complete account of the tuning can still be wrong. It is the realm of reviewers to evaluate the quality of a manuscript. But without a complete account of hyperparameter values and tuning, readers and, in particular, reviewers cannot judge whether hyperparameter tuning is technically sound.

Table 1. Can readers of a publication learn how hyperparameters were tuned and what hyperparameters were ultimately chosen? Hyperparameter explanations in papers published in APSR, PA, and PSRM between 1 January 2016 and 20 October 2021.

		Tuning Transparency	
		No	Yes
Model Transparency	No	34	2
	Yes	15	13

4. Best Practice

Hyperparameters are a fundamental element of machine learning models. Documenting their careful selection helps build trust in the insights gained from machine learning models.

4.1 Selecting Hyperparameters for Performance Tuning

Without automated procedures for finding hyperparameters, researchers need to rely on heuristics (Probst, Boulesteix, and Bischl 2018). The classic approach to hyperparameter optimization is to systematically try different hyperparameter settings and compare the models using a performance measure. Machine learning splits the data into training, validation, and test data (Friedman, Hastie, and Tibshirani 2001; Goodfellow, Bengio, and Courville 2016). The model parameters are optimized using the training data. The validation data is used to optimize the hyperparameters by estimating and then comparing an estimate of the performance of all the different models. Finally, the test data helps approximate the performance of the best model for out-of-sample data. Researchers should train a final machine learning model for a realistic estimate of the model's performance. This model relies upon the identified best set of hyperparameters, uses a combined set of the training and validation data, and is evaluated on the so far withheld test set. Note that this last evaluation can be done only once to avoid information leakage. Tuning hyperparameters is therefore not a form of "p-hacking" (Gigerenzer 2018; Wasserstein and Lazar 2016) where researchers try different models until they find the one that generates the desired statistics. On the contrary, transparently testing different hyperparameter values is necessary to find a model that generalizes well.

In hyperparameter grid search, researchers manually define a grid of hyperparameter values, then try each possible permutation and record the validation performance for each set of hyperparameters. More recently, some instead suggest randomly sampling a large number of hyperparameter candidate values from a pre-defined search space (Bergstra and Bengio 2012) and recording the validation performance of each set of sampled hyperparameter values.[5] This random search can help explore the space of hyperparameters more efficiently if some hyperparameters are more important than others. Both approaches typically yield reliable and good results for practitioners and build trust regarding the out-of-sample performance.

But the tuning of hyperparameters might be too involved for grid or random search in light of resource constraints. It is then useful to not try all combinations of hyperparameters but rather focus on the most promising ones.[6] Sequential model-based Bayesian optimization formalizes such a search for a new candidate set of hyperparameters (Shahriari et al. 2016; Snoek, Larochelle, and Adams 2012). The core idea is to formulate a surrogate model—think non-linear regression model—that predicts the machine learning model's performance for a set of hyperparameters. At iteration t, the underlying machine learning model is trained with the surrogate model's suggestion for the next best candidate set of hyperparameters. The results from this training at t are fed back into the surrogate model and used to refine the predictions for the candidate set of hyperparameters in the next iteration

5. How many permutations from the search space should be tried depends on the search space size and the available computational resources.

6. For other promising strategies, see the thorough overviews in, e.g., Bischl et al. (2021), Hutter, Lücke, and Schmidt-Thieme (2015), Luo (2016), and Probst, Boulesteix, and Bischl (2018).

$t + 1$.[7]

Without a formal solution, the selection of hyperparameters requires human judgment. We suggest relying on the following short heuristics when tuning and communicating hyperparameters.[8]

1. **Understanding the model.** What are the available hyperparameters? How do they affect the model?
2. **Choosing a performance measure.** What is a good performance for the machine learning model? Depending on the respective task, appropriate measures help assess the model's success. For example, a regression model is trained to minimize the mean squared error. Classification models can be trained to maximize the F1 score. With an appropriate performance measure, it is also possible to systematically tune the hyperparameters of unsupervised models (Fan et al. 2020).
3. **Defining a sensible search space.** Useful starting points for the hyperparameters can be the default values in software libraries, recommendations from the literature, or own previous experience (Probst, Boulesteix, and Bischl 2018). Any choice may also be informed by considerations about the data-generating process. If the hyperparameters are numerical, there may be a difference between mathematically possible and reasonable values.
4. **Finding the best combination in the search space.** In grid search, researchers should try every possible combination of the hyperparameters of the search space to find the optimal combination. In random search, each run picks a different random set of hyperparameters from the search space.
5. **Tuning under strong resource constraints.** If the model training is too involved, adaptive approaches such as sequential model-based Bayesian optimization allow for efficiently identifying and testing promising hyperparameter candidates.

Researchers should describe in either the main body or the appendix of their publication how they tuned their hyperparameters and also what final values they chose. Only then can reviewers and readers assess the robustness of machine learning models.

7. Adaptive hyperparameter optimization is conveniently implemented in many software frameworks: for R see, e.g., `mlr3` package on CRAN (Lang et al. 2019), for Python, e.g., `scikit-optimize` (Pedregosa et al. 2011) or `keras` (Chollet et al. 2015).

8. See also Bouthillier et al. (2021), Cooper et al. (2021), and Sculley et al. (2018).

4.2 Illustration: Comparing Machine Learning Models to Predict Electoral Violence from Tweets

To illustrate our point, we compare machine learning models trained to predict electoral violence from tweets. Muchlinski et al. (2021) collected Tweets around elections in three countries (Ghana, the Philippines, and Venezuela) and annotated whether these messages described occurrences of electoral violence. We re-scraped the data based on the shared Tweet IDs. To predict these occurrences from the content of these Tweets, we use four different machine learning models—a naive Bayes classifier (NB), random forest (RF), a support vector machine (SVM), and a convolutional neural network (CNN).

Table 2 summarizes our results. In the left column of each country, we report the results from training the models with default hyperparameters. On the right, we show the results after hyperparameter tuning.[9] Hyperparameter tuning improves the out-of-sample performance for most machine learning models in our experiment.[10] Table 2 also shows how easy it is to be deceived about the relative performance of different models—if hyperparameters are not properly tuned. The performance gains from tuning are so substantial that most tuned models outperform any other model with default hyperparameters. In the case of Venezuela, for example, comparing a tuned model with all other baseline models at their default hyperparameter settings could lead to different conclusions. Researchers could mistakenly conclude that (a tuned) NB classifier (F1=0.308) is better than any other method; or also that the RF is the better model (F1=0.479), or the SVM (F1=0.465), or the CNN (F1=0.298). In short, model comparisons and model choices are only meaningful if all hyperparameters of all models are systematically tuned and if this tuning is transparently documented.

Table 2. Performance benchmarking of Muchlinski et al. (2021) on different classifiers using our scraped data. On the left: results with default values for the hyperparameters. On the right: results from tuned hyperparameters.

Classifier	**Default** F1	**Tuned** F1	**Default** F1	**Tuned** F1	**Default** F1	**Tuned** F1
	Ghana		**The Philippines**		**Venezuela**	
NB	0.000	0.538	0.000	0.390	0.000	0.308
RF	0.341	0.603	0.400	0.160	0.237	0.479
SVM	0.381	0.727	0.357	0.561	0.080	0.465
CNN	0.679	0.679	0.421	0.444	0.230	0.298

9. In line with (Muchlinski et al. 2021), we chose the F1 score as the performance metric. We include details on the tuned hyperparameters, the default values we chose, the search method, the search space for each model, and any random seeds in the Appendix.

10. In cases where hyperparameter tuning does not improve the performance over default hyperparameter values, the default values are closer to the optimal solution than the best-performing hyperparameters from a cross-validation procedure. However, the only way to find this out is through systematic hyperparameter tuning.

5. Tuning Hyperparameters Matters

Hyperparameters critically influence how well machine learning models perform on unseen, out-of-sample data. Despite the relevance of tuned hyperparameters, we found that only 20.31% of the papers using machine learning models published in APSR, PA, and PSRM between 2016 and 2021 include information about the ultimate hyperparameter choice and how they were found in the manuscript or the appendix. Furthermore, 34 papers (53.12%) neither report the hyperparameters nor their tuning. This is a dangerous habit since handling hyperparameters without care can lead to wrong conclusions about model performance and model choice.

The search for an optimal set of hyperparameters is a vibrant research area in computer science and statistics. For most of the applications in our discipline, acknowledging and discussing how the choice of hyperparameters could influence results in combination with a proper and systematic search for appropriate hyperparameters would go a long way. It would allow others to understand original work, assess its validity, and thus ultimately help build trust in political science that uses machine learning.

Acknowledgement

Thomas Gschwend, Oliver Rittmann, and Zach Warner provided very insightful feedback on earlier versions of the draft. We also thank three anonymous reviewers and the editor for constructive feedback in improving the quality of the manuscript.

References

Bergstra, James, and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* 13, no. null (February): 281–305. ISSN: 1532-4435.

Bischl, Bernd, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, et al. 2021. *Hyperparameter optimization: foundations, algorithms, best practices and open challenges.* https://doi.org/10.48550/ARXIV.2107.05847. https://arxiv.org/abs/2107.05847.

Bouthillier, Xavier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Nazanin Mohammadi Sepahvand, et al. 2021. Accounting for variance in machine learning benchmarks. In *Proceedings of machine learning and systems,* edited by A. Smola, A. Dimakis, and I. Stoica, 3:747–769. https://proceedings.mlsys.org/paper/2021/file/cfecdb276f634854f3ef915e2e980c31-Paper.pdf.

Bouthillier, Xavier, César Laurent, and Pascal Vincent. 2019. Unreproducible research is reproducible. In *Proceedings of the 36th international conference on machine learning,* edited by Kamalika Chaudhuri and Ruslan Salakhutdinov, 97:725–734. Proceedings of Machine Learning Research. PMLR, September. https://proceedings.mlr.press/v97/bouthillier19a.html.

Chang, Charles, and Michael Masterson. 2020. Using word order in political text classification with long short-term memory models. *Political Analysis* 28 (3): 395–411.

Chollet, François, et al. 2015. *Keras.* https://keras.io.

Cooper, A Feder, Yucheng Lu, Jessica Forde, and Christopher M De Sa. 2021. Hyperparameter optimization is deceiving us, and how to stop it. *Advances in Neural Information Processing Systems* 34:3081–3095.

Cranmer, Skyler J, and Bruce A Desmarais. 2017. What can we learn from predictive modeling? *Political Analysis* 25 (2): 145–166.

Fan, Xinjie, Yuguang Yue, Purnamrita Sarkar, and Y. X. Rachel Wang. 2020. On hyperparameter tuning in general clustering problems. In *Proceedings of the 37th international conference on machine learning,* edited by Hal Daumé III and Aarti Singh, 119:2996–3007. Proceedings of Machine Learning Research. PMLR, 13–18 Jul. https://proceedings.mlr.press/v119/fan20b.html.

Fariss, Christopher J, and Zachary M Jones. 2018. Enhancing validity in observational settings when replication is not possible. *Political Science Research and Methods* 6 (2): 365–380.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning.* Vol. 1. 10. New York: Springer Series in Statistics.

Gigerenzer, Gerd. 2018. Statistical rituals: the replication delusion and how we got there. *Advances in Methods and Practices in Psychological Science* 1 (2): 198–218. https://doi.org/10.1177/2515245918771329. eprint: https://doi.org/10.1177/2515245918771329. https://doi.org/10.1177/2515245918771329.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning.* Http://www.deeplearningbook.org. MIT Press.

Gundersen, Odd Erik, Kevin Coakley, and Christine Kirkpatrick. 2022. *Sources of irreproducibility in machine learning: a review.* https://doi.org/10.48550/ARXIV.2204.07610. https://arxiv.org/abs/2204.07610.

Henderson, Peter, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. 2018. Deep reinforcement learning that matters. In *Proceedings of the thirty-second aaai conference on artificial intelligence and thirtieth innovative applications of artificial intelligence conference and eighth aaai symposium on educational advances in artificial intelligence.* AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA: AAAI Press. ISBN: 978-1-57735-800-8.

Hutter, Frank, Jörg Lücke, and Lars Schmidt-Thieme. 2015. Beyond manual tuning of hyperparameters. *KI-Künstliche Intelligenz* 29 (4): 329–337.

Lang, Michel, Martin Binder, Jakob Richter, Patrick Schratz, Florian Pfisterer, Stefan Coors, Quay Au, Giuseppe Casalicchio, Lars Kotthoff, and Bernd Bischl. 2019. mlr3: a modern object-oriented machine learning framework in R. *Journal of Open Source Software* (December).

Lucic, Mario, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. 2018. Are gans created equal? a large-scale study. In *Advances in neural information processing systems,* edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/file/e46de7e1bcaaced9a54f1e9d0d2f800d-Paper.pdf.

Luo, Gang. 2016. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics* 5 (1): 1–16.

Melis, Gábor, Chris Dyer, and Phil Blunsom. 2018. On the state of the art of evaluation in neural language models. In *International conference on learning representations.* https://openreview.net/forum?id=ByJHuTgA-.

Miller, Blake, Fridolin Linder, and Walter R Mebane. 2020. Active learning approaches for labeling text: review and assessment of the performance of active learning approaches. *Political Analysis* 28 (4): 532–551.

Mitchell, Tom M. 1997. *Machine learning.* McGraw-Hill International Editions. McGraw-Hill. ISBN: 9780071154673.

Muchlinski, David, Xiao Yang, Sarah Birch, Craig Macdonald, and Iadh Ounis. 2021. We need to go deeper: measuring electoral violence using convolutional neural networks and social media. *Political Science Research and Methods* 9 (1): 122–139.

Musgrave, Kevin, Serge Belongie, and Ser-Nam Lim. 2020. A metric learning reality check. In *Computer vision – eccv 2020,* edited by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, 681–699. Cham: Springer International Publishing. ISBN: 978-3-030-58595-2.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Probst, Philipp, Anne-Laure Boulesteix, and Bernd Bischl. 2018. *Tunability: importance of hyperparameters of machine learning algorithms.* arXiv: 1802.09596.

Rheault, Ludovic, and Christopher Cochrane. 2020. Word embeddings for the analysis of ideological placement in parliamentary corpora. *Political Analysis* 28 (1): 112–133.

Sculley, David, Jasper Snoek, Alex Wiltschko, and Ali Rahimi. 2018. Winner's curse? on pace, progress, and empirical rigor. *ICLR Workshop.*

Shahriari, Bobak, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. 2016. Taking the human out of the loop: a review of bayesian optimization. *Proceedings of the IEEE* 104 (1): 148–175. https://doi.org/10.1109/JPROC.2015.2494218.

Shalev-Shwartz, Shai, and Shai Ben-David. 2014. *Understanding machine learning: from theory to algorithms.* Cambridge University Press.

Snoek, Jasper, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems,* edited by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

Torres, Michelle, and Francisco Cantú. 2021. Learning to see: convolutional neural networks for the analysis of social science data. *Political Analysis,* 1–19. https://doi.org/10.1017/pan.2021.9.

Wasserstein, Ronald L., and Nicole A. Lazar. 2016. The asa statement on p-values: context, process, and purpose. *The American Statistician* 70 (2): 129–133.

Appendix 1. Collection and Coding Instructions for Papers

We scrape google scholar looking for APSR, PA, and PSRM with the search string "machine learning" in the full text of the papers after 1 January 2016 and before 20 October 2021, resulting in 137 manuscripts. We then identify those publications that use machine learning models according to our definition (Column *Applies ML?* in Table 3) For example, we exclude papers where the only mention of machine learning is in the references, e.g., in the "Journal of Machine Learning Research" or where the authors make a quick reference to machine learning approaches but do not employ machine learning themselves. Left with 65 manuscripts, we then annotate them with the following coding scheme.

- *Tunable HPs?*: Are there any tunable hyperparameters involved in the models which are described in the paper or appendix? We discard one other manuscript here (Ratkovic and Tingley 2017).
- *Model Transparency*: Are the final hyperparameter values (of all models) in the paper or appendix?
- *Tuning Transparency*: Are the hyperparameter search method (e.g., grid search) and search space (range of tested values) described in the paper or appendix?

Please allow us some further remarks concerning the annotation. First, our annotation is not a statement of the "correctness" of the approach. During the annotation process, we set the values for model and/or tuning transparency to `FALSE` for papers referencing existing work to justify their hyperparameter choice without mentioning the actual values. Furthermore, we did not check whether the authors included values for all available hyperparameters of an implementation. We assume that they use the proposed default values for the remaining hyperparameters. Next, when multiple machine learning models were used, we assigned `FALSE` to a category if one of these models failed to fulfill the requirements according to our coding scheme. Like the weakest link in a chain, the scientific rigor will be affected by the weakest part of its analysis. On several occasions, authors propose a new model, only to pitch it against a baseline from machine learning models that use default settings or even manually set values.

Appendix 2. Overview of Papers in Our Sample

Table 3 contains all 137 papers containing “machine learning” in the full text published in PSRM, PA, and APSR between 1 January 2016 and 20 October 2021. We coded 65 of these papers using machine learning models. These 65 papers are the basis of our analysis.

Table 3. Overview of all papers in our sample. We retrieved 137 papers, 65 of which applied machine learning models according to our definition. We report our coding of model transparency and tuning transparency. The symbol – indicates that our coding scheme was not applicable.

Political Science Research and Methods

Article	Applies ML?	Tunable HPs?	Model Transparency	Tuning Transparency
Settle et al. 2016	✗	-	-	-
Schutte 2017	✗	-	-	-
Bagozzi and Berliner 2018	✓	✓	✓	✓
Fariss and Jones 2018	✗	-	-	-
Wu 2018	✗	-	-	-
Hopkins and Pettingill 2018	✗	-	-	-
Munger et al. 2019	✓	✓	✓	✓
Hollenbach, Montgomery, and Crespo-Tenorio 2019	✗	-	-	-
Pan 2019	✓	✓	✗	✗
Lee, Liu, and Ward 2019	✓	✓	✗	✗
Ramey, Klingler, and Hollibaugh 2019	✓	✓	✓	✗
Kikuta 2020	✓	✓	✗	✗
Beiser-McGrath and Beiser-McGrath 2020	✓	✓	✗	✗
Baerg and Lowe 2020	✗	-	-	-
Struthers, Hare, and Bakker 2020	✗	-	-	-
Torres 2020	✗	-	-	-
Herzog and Mikhaylov 2020	✗	-	-	-
Stuckatz 2020	✗	-	-	-
Keele, Stevenson, and Elwert 2020	✗	-	-	-
Benedictis-Kessner 2020	✓	✓	✗	✗
Radford 2021	✓	✓	✓	✗
Muchlinski et al. 2021	✓	✓	✗	✗
Blaydes et al. 2021	✗	-	-	-
Rice and Zorn 2021	✗	-	-	-
Crosson 2021	✗	-	-	-

Minhas et al. 2021	✗	-	-	-
Christia et al. 2021	✗	-	-	-
Funk, Paul, and Philips 2021	✓	✓	✓	✓

Political Analysis

Article	Applies ML?	Tunable HPs?	Model Transparency	Tuning Transparency
Imai and Khanna 2016	✗	-	-	-
Kasy 2016	✗	-	-	-
Samii, Paler, and Daly 2016	✓	✓	✗	✗
Muchlinski et al. 2016	✓	✓	✗	✗
Ratkovic and Tingley 2017	✓	✗	-	-
Cranmer and Desmarais 2017	✓	✓	✗	✗
Van Atteveldt et al. 2017	✗	-	-	-
Rozenas 2017	✗	-	-	-
Tausanovitch and Warshaw 2017	✗	-	-	-
Rosenberg, Knuppe, and Braumoeller 2017	✗	-	-	-
Fafchamps and Labonne 2017	✗	-	-	-
Grimmer, Messing, and Westwood 2017	✓	✓	✗	✗
Greene and Cross 2017	✓	✓	✓	✗
De Vries, Schoonvelde, and Schumacher 2018	✓	✓	✓	✓
Denny and Spirling 2018	✓	✓	✓	✓
Kim, Londregan, and Ratkovic 2018	✗	-	-	-
Blackwell 2018	✗	-	-	-
Peterson and Spirling 2018	✓	✓	✗	✗
Temporão et al. 2018	✓	✓	✓	✗
Bansak 2019	✓	✓	✓	✗
Wang 2019	✓	✓	✗	✗
Neunhoeffer and Sternberg 2019	✓	✓	✗	✗
Kaufman, Kraft, and Sen 2019	✓	✓	✗	✗
Greene, Park, and Colaresi 2019	✓	✓	✗	✗
Goet 2019	✓	✓	✓	✓
Goplerud 2019	✗	-	-	-
Stoetzer et al. 2019	✗	-	-	-
Hainmueller, Mummolo, and Xu 2019	✗	-	-	-
De la Cuesta, Egami, and Imai 2019	✗	-	-	-
Minhas, Hoff, and Ward 2019	✗	-	-	-

Heuberger 2019	✗	-	-	-
Mohanty and Shaffer 2019	✗	-	-	-
Brandenberger 2019	✗	-	-	-
Muchlinski et al. 2019	✗	-	-	-
King and Nielsen 2019	✗	-	-	-
Jerzak, King, and Strezhnev 2019	✗	-	-	-
Miller, Linder, and Mebane 2020	✓	✓	✗	✗
Mozer et al. 2020	✓	✓	✓	✓
Ornstein 2020	✓	✓	✓	✓
Rheault and Cochrane 2020	✓	✓	✓	✗
Huang, Perry, and Spirling 2020	✗	-	-	-
Ziegler 2020	✗	-	-	-
Bølstad 2020	✗	-	-	-
Lu 2020	✗	-	-	-
Ferrari 2020	✗	-	-	-
Bussell 2020	✗	-	-	-
Rodman 2020	✓	✓	✗	✗
Marble and Tyler 2020	✗	-	-	-
Bustikova et al. 2020	✓	✓	✗	✗
Ghitza and Gelman 2020	✗	-	-	-
Lall and Robinson 2020	✓	✓	✓	✗
Chang and Masterson 2020	✓	✓	✓	✗
Duch et al. 2020	✓	✓	✗	✗
Cohen and Warner 2021	✓	✓	✗	✗
Barberá et al. 2021	✓	✓	✗	✗
Acharya, Bansak, and Hainmueller 2021	✓	✓	✗	✗
Di Cocco and Monechi 2021	✓	✓	✓	✓
Torres and Cantú 2021	✓	✓	✓	✓
Porter and Velez, n.d.	✗	-	-	-
Ying, Montgomery, and Stewart 2021	✗	-	-	-
Kaufman and Klevs 2021	✗	-	-	-
Erlich et al. 2021	✓	✓	✗	✓
Blackwell and Olson 2021	✓	✓	✗	✗
Timoneda and Wibbels 2021	✓	✓	✓	✗
Kim and Kunisky 2021	✗	-	-	-
Vannoni, Ash, and Morelli 2021	✗	-	-	-
Enamorado, López-Moctezuma, and Ratkovic 2021	✗	-	-	-
Egami 2021	✗	-	-	-

Fong and Tyler 2021	✓	✓	✗	✗
Sebők and Kacsuk 2021	✓	✓	✗	✗

American Political Science Review

Article	Applies ML?	Tunable HPs?	Model Transparency	Tuning Transparency
Benoit et al. 2016	✗	-	-	-
Rundlett and Svolik 2016	✗	-	-	-
Imai, Lo, and Olmsted 2016	✗	-	-	-
King, Pan, and Roberts 2017	✗	-	-	-
Steinert-Threlkeld 2017	✗	-	-	-
Blackwell and Glynn 2018	✗	-	-	-
Hall and Thompson 2018	✗	-	-	-
Pan and Chen 2018	✓	✓	✓	✓
Mueller and Rauh 2018	✓	✓	✓	✗
Blair et al. 2019	✗	-	-	-
Dorsch and Maarek 2019	✗	-	-	-
Hobbs and Lajevardi 2019	✗	-	-	-
Mitts 2019	✓	✓	✗	✗
Enamorado, Fifield, and Imai 2019	✗	-	-	-
Barberá et al. 2019	✓	✓	✓	✓
Bisbee 2019	✓	✓	✓	✗
Katagiri and Min 2019	✓	✓	✗	✗
Cantú 2019	✓	✓	✓	✗
Park, Greene, and Colaresi 2020	✓	✓	✗	✗
Magaloni and Rodriguez 2020	✓	✓	✓	✓
Badrinathan 2021	✗	-	-	-
Manekin and Mitts 2021	✗	-	-	-
Goel et al. 2020	✗	-	-	-
Challú, Seira, and Simpser 2020	✗	-	-	-
Nyrup and Bramwell 2020	✗	-	-	-
Yoder 2020	✓	✓	✓	✗
Peyton 2020	✓	✓	✓	✗
Anastasopoulos and Bertelli 2020	✓	✓	✗	✗
Bøggild, Aarøe, and Petersen 2021	✓	✓	✗	✗
Zubek, Dasgupta, and Doyle 2021	✓	✓	✗	✓
Jacobs et al. 2021	✓	✓	✗	✗

Bansak, Bechtel, and Margalit 2021	✓	✓	✗	✗
Knox and Lucas 2021	✗	-	-	-
Ballard and Curry 2021	✗	-	-	-
Wahman, Frantzeskakis, and Yildirim 2021	✓	✓	✓	✗
Osnabrügge, Hobolt, and Rodon 2021	✓	✓	✗	✗

Appendix 3. Details on the Machine Learning Models and Hyperparameters in the Illustration

We reanalyze Muchlinski et al. (2021) to show how hyperparameter deception may lead to wrong conclusions about machine learning models' out-of-sample performance and, with it, ultimately also model comparison. Muchlinski et al. (2021) introduce a Convolutional Neural Network (CNN) to detect electoral violence with tweets. Studying three countries (Ghana, the Philippines, and Venezuela), they compare the performance of their CNN model against a baseline from a Support Vector Machine (SVM). Re-scraping Twitter[11] based on the author's tweet IDs, we were able to access 58% of the Tweets in the Philippines, 74% of the Tweets in Venezuela, and 78% of the Tweets in Ghana. We then pre-processed the Tweets as outlined in their manuscript.

Our approach differs in three ways. First, in line with Kim (2014), who originally proposes the CNN architecture in Muchlinski et al. (2021), we find that self-learned embeddings underperform.[12] Instead, we use word embeddings for English and Spanish that have been trained on large corpora.[13] Second, we expect that machine learning models are quite sensitive in the context of medium-sized training sets. In addition to the SVM, we train a naive base classifier and a random forest classifier. Hyperparameters for those baseline models are found using grid search. Since the tuning of the CNN is more involved, we decided to implement a random search strategy for its hyperparameters.

Finally, in the main part of the paper, we report the tuning based on one single split between a 60% training set, a 20% validation set, and a 20% test set.[14] For the appendix, we implement cross-validation that avoids overfitting and generates a realistic evaluation of the generalization error across different samples (Bischl et al. 2021; Neunhoeffer and Sternberg 2019). We split our data between a 60% training set, a 20% validation set, and a 20% test set—and repeat this using different random splits three times for the resource-intensive CNN and five times for the other machine

11. In December 2020.
12. F1 scores never exceed 0.20 in any model. The rather small corpus allows observing only a limited number of word collocations.
13. English word embeddings: pretrained Google Word2Vec as in `Gensim` (Řehůřek and Sojka 2010). Spanish word embeddings: Word2Vec model trained on the Spanish Billion Words Corpus (Cardellino 2019).
14. Random seed = 20210101.

learning models. We optimize the respective machine learning model and its hyperparameters in each fold and then aggregate results across all folds.

For our performance benchmarking, we implemented five models. All models except the Convolutional Neural Network (CNN) are based on the Python-library `scikit-learn` (Pedregosa et al. 2011). For the CNN, we use `keras` (Chollet et al. 2015) as an underlying framework. The model specifications, default settings, and search ranges for the hyperparameter optimization are listed below. Additional hyperparameters not mentioned were automatically set to the default values assigned by their package implementation. In each table, we report the Tuning F1, which is calculated based on the validation set to allow for the choice of the best hyperparameters. The out-of-sample F1 score is the estimate on the test set to approximate the generalization error. Remember, knowing how well a specific hyperparameter setting will generalize to out-of-sample data is impossible in advance. Occasionally, this results in default hyperparameter values performing better on out-of-sample data than those selected after optimization on the validation set.

Naive Bayes is a probabilistic classifier based on Bayes' theorem following a strong independence assumption of tokens. We use the implementation `sklearn.naive_bayes.MultinomialNB` in the Python-library `scikit-learn` (Pedregosa et al. 2011). In this implementation, the classifier has only the hyperparameter `alpha` (Default value: 1.0). To tune this hyperparameter, we iterate over a grid search using five-fold cross-validation based on the following value range:

- `alpha`: logarithmically spaced grid from 1 to $1e-9$ with 100 steps

This means that we test 100 different hyperparameter values.

Table 4. Best Naive Bayes Hyperparameters over five seeds optimized by F1

Seed	`alpha`	Tuning F1	Out-of-Sample F1
		Ghana	
20210101	10^{-9}	0.512	0.538
20210102	10^{-9}	0.457	0.522
20210103	10^{-9}	0.452	0.415
20210104	10^{-9}	0.444	0.632
20210105	10^{-9}	0.456	0.468
		The Philippines	
20210101	10^{-9}	0.482	0.390
20210102	10^{-9}	0.449	0.421
20210103	10^{-9}	0.465	0.324
20210104	10^{-9}	0.448	0.474
20210105	10^{-9}	0.462	0.526
		Venezuela	
20210101	0.002	0.331	0.308
20210102	0.002	0.321	0.358
20210103	0.004	0.347	0.344
20210104	0.019	0.290	0.480
20210105	0.004	0.340	0.333

Random Forest is a classifier based on an ensemble of decision trees that are fitted on sub-samples of the training dataset. It was introduced by Breiman 2001. We use the implementation `sklearn.ensemble.RandomForestClassifier` in the Python-library `scikit-learn` (Pedregosa et al. 2011). In this implementation, the classifier has a wide range of hyperparameters. A selection of them are `n_estimators` (Default value: 100), `criterion` (Default value: gini), `max_depth` (Default value: None), `max_features` (Default value: sqrt) and `class_weight` (Default value: None). We tune these hyperparameters while keeping the implementations' default values for the remainder. To optimize the hyperparameters of our RFs, we iterate over a grid search using five-fold cross-validation based on the following range of values:

- `n_estimators`: 1, 5, 15, 50, 75, 100, 150, 200, 400, 1000
- `max_depth`: 1, 5, 25, 50, 75, 100, 150, 200, 400, 1000, None
- `max_features`: sqrt, log2, None
- `class_weight`: balanced, None

This means we test a total of $10 \times 11 \times 3 \times 2 = 660$ different permutations of hyperparameter values.

Table 5. Best Random Forest Hyperparameters over five seeds optimized by F1

Seed	n_estimators	max_depth	max_features	class_weight	Tuning F1	Out-of-Sample F1
			Ghana			
20210101	100	5	sqrt	balanced	0.599	0.603
20210102	200	5	sqrt	balanced	0.592	0.472
20210103	150	5	sqrt	balanced	0.611	0.551
20210104	150	5	sqrt	balanced	0.581	0.500
20210105	400	5	sqrt	balanced	0.597	0.545
			The Philippines			
20210101	400	1	log2	balanced	0.462	0.160
20210102	1000	5	sqrt	balanced	0.472	0.417
20210103	1000	5	log2	balanced	0.517	0.256
20210104	150	5	sqrt	balanced	0.459	0.458
20210105	100	5	sqrt	balanced	0.466	0.372
			Venezuela			
20210101	1000	5	sqrt	balanced	0.486	0.479
20210102	150	5	sqrt	balanced	0.505	0.283
20210103	400	5	sqrt	balanced	0.469	0.516
20210104	400	5	sqrt	balanced	0.486	0.491
20210105	200	5	sqrt	balanced	0.480	0.420

A **Support Vector Machine** is an algorithm that finds a hyperplane to maximize the separation between different classes. The idea of support vectors was first introduced by Boser, Guyon, and Vapnik 1992. We use the implementation `sklearn.svm.SVC` in the Python-library `scikit-learn` (Pedregosa et al. 2011). Again, this implementation offers a wide range of hyperparameters. A selection of them are `C` (Default value: 1), `kernel` (Default value: rbf), `gamma` (Default value: scale) and `class_weight` (Default value: None). We tune these hyperparameters while keeping the implementations' default values for the remainder. To optimize them, we iterate over a grid search using five-fold cross-validation based on the following range of values:

- `C`: $\exp\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
- `kernel`: linear, rbf, poly, sigmoid
- `gamma`: (applies only if the kernel is not linear, otherwise None) 0.0001, 0.001, 0.01, 0.1, 1, scale, auto
- `class_weight`: balanced, None

This means we test a total of $11 \times 3 \times 7 \times 2 + 11 \times 2 = 484$ permutations of hyperparameter values.

Table 6. Best Support Vector Machine Hyperparameters over five seeds optimized by F1

Seed	`C`	`kernel`	`gamma`	`class_weight`	Tuning F1	Out-of-Sample F1
			Ghana			
20210101	20.086	rbf	0.01	balanced	0.674	0.727
20210102	2980.958	rbf	0.0001	balanced	0.666	0.597
20210103	2.718	sigmoid	0.1	balanced	0.657	0.595
20210104	148.413	rbf	0.001	balanced	0.671	0.560
20210105	20.086	sigmoid	0.01	balanced	0.684	0.640
			The Philippines			
20210101	2980.958	rbf	log2	balanced	0.521	0.561
20210102	148.413	rbf	sqrt	None	0.551	0.424
20210103	2980.958	sigmoid	log2	None	0.569	0.488
20210104	20.086	rbf	sqrt	balanced	0.547	0.542
20210105	20.086	rbf	sqrt	balanced	0.550	0.512
			Venezuela			
20210101	1.0	rbf	0.1	balanced	0.538	0.465
20210102	403.429	rbf	0.0001	balanced	0.541	0.446
20210103	1.0	rbf	0.01	balanced	0.558	0.500
20210104	148.413	rbf	auto	balanced	0.499	0.531
20210105	54.598	sigmoid	0.001	balanced	0.527	0.547

A **Convolutional Neural Network** is a deep learning algorithm primarily used for the classification of images but also text. Modern CNNs for image classification were introduced by Cun et al. 1990, and we use the implementation offered by the Python framework `keras` (Chollet et al. 2015). As this implementation offers a wide range of hyperparameters, we focus on a selection of them. These are the number of `filters` (Default value: 200), `kernel size` (Default value: 1), `dropout` probability (Default value: 0.5), `L2 regularization` (Default value: 0.01) and `learning rate` (Default value: 0.001). We tune these hyperparameters while keeping the implementations' default values for the remainder. To optimize the hyperparameters of our CNN, we iterate over 50 random combinations of parameters in each fold of a three-fold cross-validation. These parameter combinations are based on the following range of values:

- `filters`: 150, 200, 250
- `kernel size`: [1,2,3], [2,3,4], [3,4,5]
- `dropout`: 0.5, 0.8
- `L2 regularization`: 0.001, 0.01, 0.1
- `learning rate`: 0.01, 0.001, 0.0001

This means we test 50 randomly chosen permutations of hyperparameters out of $3 \times 3 \times 2 \times 3 \times 3 = 162$ possible permutations.

Table 7. Best Convolutional Neural Network Hyperparameters over three seeds optimized by AUC

Seed	`filters`	`kernel size`	`dropout`	`L2 regularization`	`learning rate`	Out-of-Sample F1
			Ghana			
20210101	150	[1,2,3]	0.5	0.01	0.001	0.679
20210102	150	[1,2,3]	0.5	0.001	0.0001	0.646
20210103	200	[3,4,5]	0.5	0.01	0.001	0.575
			The Philippines			
20210101	250	[2,3,4]	0.5	0.001	0.0001	0.444
20210102	200	[2,3,4]	0.5	0.001	0.0001	0.488
20210103	250	[2,3,4]	0.5	0.001	0.0001	0.304
			Venezuela			
20210101	250	[2,3,4]	0.5	0.001	0.0001	0.298
20210102	250	[2,3,4]	0.5	0.001	0.0001	0.385
20210103	200	[2,3,4]	0.5	0.001	0.0001	0.390

References

Acharya, Avidit, Kirk Bansak, and Jens Hainmueller. 2021. Combining outcome-based and preference-based matching: a constrained priority mechanism. *Political Analysis,* 1–24.

Anastasopoulos, L Jason, and Anthony M Bertelli. 2020. Understanding delegation through machine learning: a method and application to the european union. *American Political Science Review* 114 (1): 291–301.

Badrinathan, Sumitra. 2021. Educative interventions to combat misinformation: evidence from a field experiment in india. *American Political Science Review* 115 (4): 1325–1341.

Baerg, Nicole, and Will Lowe. 2020. A textual taylor rule: estimating central bank preferences combining topic and scaling methods. *Political Science Research and Methods* 8 (1): 106–122.

Bagozzi, Benjamin E, and Daniel Berliner. 2018. The politics of scrutiny in human rights monitoring: evidence from structural topic models of us state department human rights reports. *Political Science Research and Methods* 6 (4): 661–677.

Ballard, Andrew O, and James M Curry. 2021. Minority party capacity in congress. *American Political Science Review,* 1–18.

Bansak, Kirk. 2019. Can nonexperts really emulate statistical learning methods? a comment on "the accuracy, fairness, and limits of predicting recidivism". *Political Analysis* 27 (3): 370–380.

Bansak, Kirk, Michael M Bechtel, and Yotam Margalit. 2021. Why austerity? the mass politics of a contested policy. *American Political Science Review* 115 (2): 486–505.

Barberá, Pablo, Amber E Boydstun, Suzanna Linn, Ryan McMahon, and Jonathan Nagler. 2021. Automated text classification of news articles: a practical guide. *Political Analysis* 29 (1): 19–42.

Barberá, Pablo, Andreu Casas, Jonathan Nagler, Patrick J Egan, Richard Bonneau, John T Jost, and Joshua A Tucker. 2019. Who leads? who follows? measuring issue attention and agenda setting by legislators and the mass public using social media data. *American Political Science Review* 113 (4): 883–901.

Beiser-McGrath, Janina, and Liam F Beiser-McGrath. 2020. Problems with products? control strategies for models with interaction and quadratic effects. *Political Science Research and Methods* 8 (4): 707–730.

Benedictis-Kessner, Justin de. 2020. Strategic government communication about performance. *Political Science Research and Methods,* 1–16.

Benoit, Kenneth, Drew Conway, Benjamin E Lauderdale, Michael Laver, and Slava Mikhaylov. 2016. Crowd-sourced text analysis: reproducible and agile production of political data. *American Political Science Review* 110 (2): 278–295.

Bisbee, James. 2019. Barp: improving mister p using bayesian additive regression trees. *American Political Science Review* 113 (4): 1060–1065.

Bischl, Bernd, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, et al. 2021. *Hyperparameter optimization: foundations, algorithms, best practices and open challenges.* https://doi.org/10.48550/ARXIV.2107.05847. https://arxiv.org/abs/2107.05847.

Blackwell, Matthew. 2018. Game changers: detecting shifts in overdispersed count data. *Political Analysis* 26 (2): 230–239.

Blackwell, Matthew, and Adam N Glynn. 2018. How to make causal inferences with time-series cross-sectional data under selection on observables. *American Political Science Review* 112 (4): 1067–1082.

Blackwell, Matthew, and Michael P Olson. 2021. Reducing model misspecification and bias in the estimation of interactions. *Political Analysis,* 1–20.

Blair, Graeme, Jasper Cooper, Alexander Coppock, and Macartan Humphreys. 2019. Declaring and diagnosing research designs. *American Political Science Review* 113 (3): 838–859.

Blaydes, Lisa, et al. 2021. Authoritarian media and diversionary threats: lessons from 30 years of syrian state discourse. *Political Science Research and Methods* 9 (4): 693–708.

Bøggild, Troels, Lene Aarøe, and Michael Bang Petersen. 2021. Citizens as complicits: distrust in politicians and biased social dissemination of political information. *American Political Science Review* 115 (1): 269–285.

Bølstad, Jørgen. 2020. Capturing rationalization bias and differential item functioning: a unified bayesian scaling approach. *Political Analysis* 28 (3): 340–355.

Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory,* 144–152. COLT '92. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery. ISBN: 089791497X. https://doi.org/10.1145/130385.130401. https://doi.org/10.1145/130385.130401.

Brandenberger, Laurence. 2019. Predicting network events to assess goodness of fit of relational event models. *Political Analysis* 27 (4): 556–571.

Breiman, Leo. 2001. Random forests. *Mach. Learn.* (USA) 45, no. 1 (October): 5–32. ISSN: 0885-6125. https://doi.org/10.1023/A:1010933404324. https://doi.org/10.1023/A:1010933404324.

Bussell, Jennifer. 2020. Shadowing as a tool for studying political elites. *Political Analysis* 28 (4): 469–486.

Bustikova, Lenka, David S Siroky, Saud Alashri, and Sultan Alzahrani. 2020. Predicting partisan responsiveness: a probabilistic text mining time-series approach. *Political Analysis* 28 (1): 47–64.

Cantú, Francisco. 2019. The fingerprints of fraud: evidence from mexico's 1988 presidential election. *American Political Science Review* 113 (3): 710–726.

Cardellino, Cristian. 2019. *Spanish Billion Words Corpus and Embeddings.* https://crscardellino.github.io/SBWCE/.

Challú, Cristian, Enrique Seira, and Alberto Simpser. 2020. The quality of vote tallies: causes and consequences. *American Political Science Review* 114 (4): 1071–1085.

Chang, Charles, and Michael Masterson. 2020. Using word order in political text classification with long short-term memory models. *Political Analysis* 28 (3): 395–411.

Chollet, François, et al. 2015. *Keras.* https://keras.io.

Christia, Fotini, Spyros I Zoumpoulis, Michael Freedman, Leon Yao, and Ali Jadbabaie. 2021. The effect of drone strikes on civilian communication: evidence from yemen. *Political Science Research and Methods,* 1–9.

Cohen, Mollie J, and Zach Warner. 2021. How to get better survey data more efficiently. *Political Analysis* 29 (2): 121–138.

Cranmer, Skyler J, and Bruce A Desmarais. 2017. What can we learn from predictive modeling? *Political Analysis* 25 (2): 145–166.

Crosson, Jesse. 2021. Extreme districts, moderate winners: same-party challenges, and deterrence in top-two primaries. *Political science research and methods* 9 (3): 532–548.

Cun, Y. Le, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. 1990. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems 2,* 396–404. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 1558601007.

De la Cuesta, Brandon, Naoki Egami, and Kosuke Imai. 2019. Improving the external validity of conjoint analysis: the essential role of profile distribution. *Political Analysis,* 1–27.

De Vries, Erik, Martijn Schoonvelde, and Gijs Schumacher. 2018. No longer lost in translation: evidence that google translate works for comparative bag-of-words text applications. *Political Analysis* 26 (4): 417–430.

Denny, Matthew J, and Arthur Spirling. 2018. Text preprocessing for unsupervised learning: why it matters, when it misleads, and what to do about it. *Political Analysis* 26 (2): 168–189.

Di Cocco, Jessica, and Bernardo Monechi. 2021. How populist are parties? measuring degrees of populism in party manifestos using supervised machine learning. *Political Analysis,* 1–17.

Dorsch, Michael T, and Paul Maarek. 2019. Democratization and the conditional dynamics of income distribution. *American Political Science Review* 113 (2): 385–404.

Duch, Raymond, Denise Laroze, Thomas Robinson, and Pablo Beramendi. 2020. Multi-modes for detecting experimental measurement error. *Political Analysis* 28 (2): 263–283.

Egami, Naoki. 2021. Spillover effects in the presence of unobserved networks. *Political Analysis,* 1–30.

Enamorado, Ted, Benjamin Fifield, and Kosuke Imai. 2019. Using a probabilistic model to assist merging of large-scale administrative records. *American Political Science Review* 113 (2): 353–371.

Enamorado, Ted, Gabriel López-Moctezuma, and Marc Ratkovic. 2021. Scaling data from multiple sources. *Political Analysis* 29 (2): 212–235.

Erlich, Aaron, Stefano G Dantas, Benjamin E Bagozzi, Daniel Berliner, and Brian Palmer-Rubin. 2021. Multi-label prediction for political text-as-data. *Political Analysis,* 1–18.

Fafchamps, Marcel, and Julien Labonne. 2017. Using split samples to improve inference on causal effects. *Political Analysis* 25 (4): 465–482.

Fariss, Christopher J, and Zachary M Jones. 2018. Enhancing validity in observational settings when replication is not possible. *Political Science Research and Methods* 6 (2): 365–380.

Ferrari, Diogo. 2020. Modeling context-dependent latent effect heterogeneity. *Political Analysis* 28 (1): 20–46.

Fong, Christian, and Matthew Tyler. 2021. Machine learning predictions as regression covariates. *Political Analysis* 29 (4): 467–484.

Funk, Kendall D, Hannah L Paul, and Andrew Q Philips. 2021. Point break: using machine learning to uncover a critical mass in women's representation. *Political Science Research and Methods,* 1–19.

Ghitza, Yair, and Andrew Gelman. 2020. Voter registration databases and mrp: toward the use of large-scale databases in public opinion research. *Political Analysis* 28 (4): 507–531.

Goel, Sharad, Marc Meredith, Michael Morse, David Rothschild, and Houshmand Shirani-Mehr. 2020. One person, one vote: estimating the prevalence of double voting in us presidential elections. *American Political Science Review* 114 (2): 456–469.

Goet, Niels D. 2019. Measuring polarization with text analysis: evidence from the uk house of commons, 1811–2015. *Political Analysis* 27 (4): 518–539.

Goplerud, Max. 2019. A multinomial framework for ideal point estimation. *Political Analysis* 27 (1): 69–89.

Greene, Derek, and James P Cross. 2017. Exploring the political agenda of the european parliament using a dynamic topic modeling approach. *Political Analysis* 25 (1): 77–94.

Greene, Kevin T, Baekkwan Park, and Michael Colaresi. 2019. Machine learning human rights and wrongs: how the successes and failures of supervised learning algorithms can inform the debate about information effects. *Political Analysis* 27 (2): 223–230.

Grimmer, Justin, Solomon Messing, and Sean J Westwood. 2017. Estimating heterogeneous treatment effects and the effects of heterogeneous treatments with ensemble methods. *Political Analysis* 25 (4): 413–434.

Hainmueller, Jens, Jonathan Mummolo, and Yiqing Xu. 2019. How much should we trust estimates from multiplicative interaction models? simple tools to improve empirical practice. *Political Analysis* 27 (2): 163–192.

Hall, Andrew B, and Daniel M Thompson. 2018. Who punishes extremist nominees? candidate ideology and turning out the base in us elections. *American Political Science Review* 112 (3): 509–524.

Herzog, Alexander, and Slava Jankin Mikhaylov. 2020. Intra-cabinet politics and fiscal governance in times of austerity. *Political Science Research and Methods* 8 (3): 409–424.

Heuberger, Simon. 2019. Insufficiencies in data material: a replication analysis of muchlinski, siroky, he, and kocher (2016). *Political Analysis* 27 (1): 114–118.

Hobbs, William, and Nazita Lajevardi. 2019. Effects of divisive political campaigns on the day-to-day segregation of arab and muslim americans. *American Political Science Review* 113 (1): 270–276.

Hollenbach, Florian M, Jacob M Montgomery, and Adriana Crespo-Tenorio. 2019. Bayesian versus maximum likelihood estimation of treatment effects in bivariate probit instrumental variable models. *Political Science Research and Methods* 7 (3): 651–659.

Hopkins, Daniel J, and Lindsay M Pettingill. 2018. Retrospective voting in big-city us mayoral elections. *Political Science Research and Methods* 6 (4): 697–714.

Huang, Leslie, Patrick O Perry, and Arthur Spirling. 2020. A general model of author "style" with application to the uk house of commons, 1935–2018. *Political Analysis* 28 (3): 412–434.

Imai, Kosuke, and Kabir Khanna. 2016. Improving ecological inference by predicting individual ethnicity from voter registration records. *Political Analysis* 24 (2): 263–272.

Imai, Kosuke, James Lo, and Jonathan Olmsted. 2016. Fast estimation of ideal points with massive data. *American Political Science Review* 110 (4): 631–656.

Jacobs, Alan M, J Scott Matthews, Timothy Hicks, and Eric Merkley. 2021. Whose news? class-biased economic reporting in the united states. *American Political Science Review,* 1–18.

Jerzak, Connor T, Gary King, and Anton Strezhnev. 2019. An improved method of automated nonparametric content analysis for social science. *Political Analysis.*

Kasy, Maximilian. 2016. Why experimenters might not always want to randomize, and what they could do instead. *Political Analysis* 24 (3): 324–338.

Katagiri, Azusa, and Eric Min. 2019. The credibility of public and private signals: a document-based approach. *American Political Science Review* 113 (1): 156–172.

Kaufman, Aaron R, and Aja Klevs. 2021. Adaptive fuzzy string matching: how to merge datasets with only one (messy) identifying field. *Political Analysis,* 1–7.

Kaufman, Aaron Russell, Peter Kraft, and Maya Sen. 2019. Improving supreme court forecasting using boosted decision trees. *Political Analysis* 27 (3): 381–387.

Keele, Luke, Randolph T Stevenson, and Felix Elwert. 2020. The causal interpretation of estimated associations in regression models. *Political Science Research and Methods* 8 (1): 1–13.

Kikuta, Kyosuke. 2020. A new geography of civil war: a machine learning approach to measuring the zones of armed conflicts. *Political Science Research and Methods,* 1–19.

Kim, In Song, and Dmitriy Kunisky. 2021. Mapping political communities: a statistical analysis of lobbying networks in legislative politics. *Political Analysis* 29 (3): 317–336.

Kim, In Song, John Londregan, and Marc Ratkovic. 2018. Estimating spatial preferences from votes and text. *Political Analysis* 26 (2): 210–229.

Kim, Yoon. 2014. *Convolutional neural networks for sentence classification.* https://doi.org/10.48550/ARXIV.1408.5882. https://arxiv.org/abs/1408.5882.

King, Gary, and Richard Nielsen. 2019. Why propensity scores should not be used for matching. *Political Analysis* 27 (4): 435–454.

King, Gary, Jennifer Pan, and Margaret E Roberts. 2017. How the chinese government fabricates social media posts for strategic distraction, not engaged argument. *American political science review* 111 (3): 484–501.

Knox, Dean, and Christopher Lucas. 2021. A dynamic model of speech for the social sciences. *American Political Science Review* 115 (2): 649–666.

Lall, Ranjit, and Thomas Robinson. 2020. The midas touch: accurate and scalable missing-data imputation with deep learning. *Political Analysis,* 1–18.

Lee, Sophie J, Howard Liu, and Michael D Ward. 2019. Lost in space: geolocation in event data. *Political science research and methods* 7 (4): 871–888.

Lu, Xiao. 2020. Discrete choice data with unobserved heterogeneity: a conditional binary quantile model. *Political Analysis* 28 (2): 147–167.

Magaloni, Beatriz, and Luis Rodriguez. 2020. Institutionalized police brutality: torture, the militarization of security, and the reform of inquisitorial criminal justice in mexico. *American Political Science Review* 114 (4): 1013–1034.

Manekin, Devorah, and Tamar Mitts. 2021. Effective for whom? ethnic identity and nonviolent resistance. *American Political Science Review,* 1–20.

Marble, William, and Matthew Tyler. 2020. The structure of political choices: distinguishing between constraint and multidimensionality. *Political Analysis,* 1–18.

Miller, Blake, Fridolin Linder, and Walter R Mebane. 2020. Active learning approaches for labeling text: review and assessment of the performance of active learning approaches. *Political Analysis* 28 (4): 532–551.

Minhas, Shahryar, Cassy Dorff, Max B Gallop, Margaret Foster, Howard Liu, Juan Tellez, and Michael D Ward. 2021. Taking dyads seriously. *Political Science Research and Methods,* 1–19.

Minhas, Shahryar, Peter D Hoff, and Michael D Ward. 2019. Inferential approaches for network analysis: amen for latent factor models. *Political Analysis* 27 (2): 208–222.

Mitts, Tamar. 2019. From isolation to radicalization: anti-muslim hostility and support for isis in the west. *American Political Science Review* 113 (1): 173–194.

Mohanty, Pete, and Robert Shaffer. 2019. Messy data, robust inference? navigating obstacles to inference with bigkrls. *Political Analysis* 27 (2): 127–144.

Mozer, Reagan, Luke Miratrix, Aaron Russell Kaufman, and L Jason Anastasopoulos. 2020. Matching with text data: an experimental evaluation of methods for matching documents and of measuring match quality. *Political Analysis* 28 (4): 445–468.

Muchlinski, David, David Siroky, Jingrui He, and Matthew Kocher. 2016. Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data. *Political Analysis* 24 (1): 87–103.

Muchlinski, David, David Siroky, Jingrui He, and Matthew Adam Kocher. 2019. Seeing the forest through the trees. *Political Analysis* 27 (1): 111–113.

Muchlinski, David, Xiao Yang, Sarah Birch, Craig Macdonald, and Iadh Ounis. 2021. We need to go deeper: measuring electoral violence using convolutional neural networks and social media. *Political Science Research and Methods* 9 (1): 122–139.

Mueller, Hannes, and Christopher Rauh. 2018. Reading between the lines: prediction of political violence using newspaper text. *American Political Science Review* 112 (2): 358–375.

Munger, Kevin, Richard Bonneau, Jonathan Nagler, and Joshua A Tucker. 2019. Elites tweet to get feet off the streets: measuring regime social media strategies during protest. *Political Science Research and Methods* 7 (4): 815–834.

Neunhoeffer, Marcel, and Sebastian Sternberg. 2019. How cross-validation can go wrong and what to do about it. *Political Analysis* 27 (1): 101–106.

Nyrup, Jacob, and Stuart Bramwell. 2020. Who governs? a new global dataset on members of cabinets. *American Political Science Review* 114 (4): 1366–1374.

Ornstein, Joseph T. 2020. Stacked regression and poststratification. *Political Analysis* 28 (2): 293–301.

Osnabrügge, Moritz, Sara B Hobolt, and Toni Rodon. 2021. Playing to the gallery: emotive rhetoric in parliaments. *American Political Science Review,* 1–15.

Pan, Jennifer. 2019. How chinese officials use the internet to construct their public image. *Political Science Research and Methods* 7 (2): 197–213.

Pan, Jennifer, and Kaiping Chen. 2018. Concealing corruption: how chinese officials distort upward reporting of online grievances. *American Political Science Review* 112 (3): 602–620.

Park, Baekkwan, Kevin Greene, and Michael Colaresi. 2020. Human rights are (increasingly) plural: learning the changing taxonomy of human rights from large-scale text reveals information effects. *American Political Science Review* 114 (3): 888–910.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Peterson, Andrew, and Arthur Spirling. 2018. Classification accuracy as a substantive quantity of interest: measuring polarization in westminster systems. *Political Analysis* 26 (1): 120–128.

Peyton, Kyle. 2020. Does trust in government increase support for redistribution? evidence from randomized survey experiments. *American Political Science Review* 114 (2): 596–602.

Porter, Ethan, and Yamil R Velez. n.d. Placebo selection in survey experiments: an agnostic approach. *Political Analysis,* 1–14.

Radford, Benjamin J. 2021. Automated dictionary generation for political eventcoding. *Political Science Research and Methods* 9 (1): 157–171.

Ramey, Adam J, Jonathan D Klingler, and Gary E Hollibaugh. 2019. Measuring elite personality using speech. *Political Science Research and Methods* 7 (1): 163–184.

Ratkovic, Marc, and Dustin Tingley. 2017. Sparse estimation and uncertainty with application to subgroup analysis. *Political Analysis* 25 (1): 1–40.

Řehůřek, Radim, and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks,* 45–50. Valletta, Malta: ELRA, May.

Rheault, Ludovic, and Christopher Cochrane. 2020. Word embeddings for the analysis of ideological placement in parliamentary corpora. *Political Analysis* 28 (1): 112–133.

Rice, Douglas R, and Christopher Zorn. 2021. Corpus-based dictionaries for sentiment analysis of specialized vocabularies. *Political Science Research and Methods* 9 (1): 20–35.

Rodman, Emma. 2020. A timely intervention: tracking the changing meanings of political concepts with word vectors. *Political Analysis* 28 (1): 87–111.

Rosenberg, Andrew S, Austin J Knuppe, and Bear F Braumoeller. 2017. Unifying the study of asymmetric hypotheses. *Political Analysis* 25 (3): 381–401.

Rozenas, Arturas. 2017. Detecting election fraud from irregularities in vote-share distributions. *Political Analysis* 25 (1): 41–56.

Rundlett, Ashlea, and Milan W Svolik. 2016. Deliver the vote! micromotives and macrobehavior in electoral fraud. *American Political Science Review* 110 (1): 180–197.

Samii, Cyrus, Laura Paler, and Sarah Zukerman Daly. 2016. Retrospective causal inference with machine learning ensembles: an application to anti-recidivism policies in colombia. *Political Analysis* 24 (4): 434–456.

Schutte, Sebastian. 2017. Regions at risk: predicting conflict zones in african insurgencies. *Political Science Research and Methods* 5 (3): 447–465.

Sebők, Miklós, and Zoltán Kacsuk. 2021. The multiclass classification of newspaper articles with machine learning: the hybrid binary snowball approach. *Political Analysis* 29 (2): 236–249.

Settle, Jaime E, Robert M Bond, Lorenzo Coviello, Christopher J Fariss, James H Fowler, and Jason J Jones. 2016. From posting to voting: the effects of political competition on online political engagement. *Political Science Research and Methods* 4 (2): 361–378.

Steinert-Threlkeld, Zachary C. 2017. Spontaneous collective action: peripheral mobilization during the arab spring. *American Political Science Review* 111 (2): 379–403.

Stoetzer, Lukas F, Marcel Neunhoeffer, Thomas Gschwend, Simon Munzert, and Sebastian Sternberg. 2019. Forecasting elections in multiparty systems: a bayesian approach combining polls and fundamentals. *Political Analysis* 27 (2): 255–262.

Struthers, Cory L, Christopher Hare, and Ryan Bakker. 2020. Bridging the pond: measuring policy positions in the united states and europe. *Political Science Research and Methods* 8 (4): 677–691.

Stuckatz, Jan. 2020. Political alignment between firms and employees in the united states: evidence from a new dataset. *Political Science Research and Methods,* 1–11.

Tausanovitch, Chris, and Christopher Warshaw. 2017. Estimating candidates' political orientation in a polarized congress. *Political Analysis* 25 (2): 167–187.

Temporão, Mickael, Corentin Vande Kerckhove, Clifton van der Linden, Yannick Dufresne, and Julien M Hendrickx. 2018. Ideological scaling of social media users: a dynamic lexicon approach. *Political Analysis* 26 (4): 457–473.

Timoneda, Joan C, and Erik Wibbels. 2021. Spikes and variance: using google trends to detect and forecast protests. *Political Analysis,* 1–18.

Torres, Michelle. 2020. Estimating controlled direct effects through marginal structural models. *Political Science Research and Methods* 8 (3): 391–408.

Torres, Michelle, and Francisco Cantú. 2021. Learning to see: convolutional neural networks for the analysis of social science data. *Political Analysis,* 1–19.

Van Atteveldt, Wouter, Tamir Sheafer, Shaul R Shenhav, and Yair Fogel-Dror. 2017. Clause analysis: using syntactic information to automatically extract source, subject, and predicate from texts with an application to the 2008–2009 gaza war. *Political Analysis* 25 (2): 207–222.

Vannoni, Matia, Elliott Ash, and Massimo Morelli. 2021. Measuring discretion and delegation in legislative texts: methods and application to us states. *Political Analysis* 29 (1): 43–57.

Wahman, Michael, Nikolaos Frantzeskakis, and Tevfik Murat Yildirim. 2021. From thin to thick representation: how a female president shapes female parliamentary behavior. *American Political Science Review* 115 (2): 360–378.

Wang, Yu. 2019. Comparing random forest with logistic regression for predicting class-imbalanced civil war onset data: a comment. *Political Analysis* 27 (1): 107–110.

Wu, Nicole. 2018. Misattributed blame? attitudes toward globalization in the age of automation. *Political Science Research and Methods,* 1–18.

Ying, Luwei, Jacob M Montgomery, and Brandon M Stewart. 2021. Topics, concepts, and measurement: a crowdsourced procedure for validating topics as measures. *Political Analysis,* 1–20.

Yoder, Jesse. 2020. Does property ownership lead to participation in local politics? evidence from property records and meeting minutes. *American Political Science Review* 114 (4): 1213–1229.

Ziegler, Jeffrey. 2020. A text-as-data approach for using open-ended responses as manipulation checks. *Political Analysis,* 1–9.

Zubek, Radoslaw, Abhishek Dasgupta, and David Doyle. 2021. Measuring the significance of policy outputs with positive unlabeled learning. *American Political Science Review* 115 (1): 339–346.