Model Info Sheet

**Section 1: Information about paper or report**

1) Author(s): Names of the authors of the paper or report

Hounsel et al. (2020)

2) Title of the paper or report which introduces the model

*Identifying Disinformation Websites Using Infrastructure Features*

3) DOI or permanent link to the paper or report (for example, link to arxiv.org webpage)

<https://www.usenix.org/system/files/foci20-paper-hounsel.pdf>

4) License: Under which license(s) are the data and/or model shared?

5) Email address of the corresponding author

*slack@mayerware.com*

**Section 2: Scientific claim(s) of interest**

6) Does your paper make a generalizable claim based on the ML model? If yes, what is the scientific claim? For example, “Our ML model can be used to diagnose Covid-19 using chest radiographs of adult patients”.

*The authors claim that they’ve described a set of 30+ infrastructure features which provide an asymmetric advantage in the early detection of misinformation. Though they refrain from making any strong claims about the generalizability of their methods, the authors do state that their methods are comparable to the state-of-the-art in website classification (in an experimental setting, at least) and concede that classifier performance degrades in real-life deployment settings.*

7) Is the scientific claim made about a distribution or population from which you can sample? If yes: (a) what is the population or distribution about which the scientific claim is being made? (b) What is the sample used for the study? For example, “(a) Population: adult patients with symptoms of Covid-19. (b) Sample: We use a random sample of adult patients who present at a U.S. based hospital between April 2020 and June 2020”.

*The authors constructed initial disinfo, news, and non-news datasets as follows:*

* *disinfo website dataset was constructed from a combination of Snopes, CBS, FactCheck.org, Wiki, BuzzFeed, and Politifact, with filtering to select for sites that aligned with the researchers’ defn of ‘disinfo’*
* *news website dataset was constructed from Amazon’s Alexa Web Information Service directory, and from a directory of local news media (websites, magazines, etc) [Pink slime journalism; see* [*here*](https://www.cjr.org/tow_center_reports/hundreds-of-pink-slime-local-news-outlets-are-distributing-algorithmic-stories-conservative-talking-points.php)*]*
* *non-news website dataset was constructed via sampling of Twitter’s Streaming API.*

*For a pilot real-time deployment study, the authors used a commodity server to ingest new domains from a number of services (DomainTools, CertStream, Twitter Streaming and Reddit APIs) that log new domain registrations.*

* *Using these tools, the authors classified 1m + websites.*
* *To simulate classification of sites on a social media site with access to all infrastructure features, the authors randomly sampled 300 sites that appeared on Twitter [why Twitter in particular? – want to sample from long tail; API readily available and easy-to-use], 100 each in the three study domains (disinformation sites, news sites, non-news sites).*

8) Does the scientific claim only apply to certain subsets of the distribution mentioned in Q6? For example, “Our model works on chest radiographs of U.S.-based adult patients and might not generalize to radiographs taken in other places or using different machines.”

*The authors state that their classifier matches state-of-the-art performance on datasets extracted from vetted sources. This performance degrades ~~slightly~~ significantly on real-time test sets.*

*‘sufficient for deployment’: surfaced a few novel disinfo sites, this is progress!*

**Section 3: Train-test split is maintained across all steps in creating the model**

9) Train-test split type: How was the dataset split into train and test sets? (For example, cross-validation; separate train and test sets).

*This is a bit murky to me – the authors mention that they designed their curated dataset to have three categories of comparable size in order to ensure balanced testing/training. They don’t, however, describe how they actually train their classifier (only that they use five-fold cross-validation after the fact to tune hyperparameters).*

10) Are there duplicates in the dataset? If yes, explain how duplicates are handled to ensure the train-test split.

*All websites within test categories (disinfo, news, non-news) are unique. Additionally, these categories are mutually exclusive by construction (impossible for a site to belong to e.g. both news* and *non-news), so no duplicates exist across categories.*

11) In case the dataset has dependencies (e.g., multiple rows of data from the same patient), describe how the dependencies were addressed (for example, using block-cross validation).

*No dependencies to flag!*

12) List all the pre-processing steps used in creating your model. For example, imputing missing data, normalizing feature values, selecting a subset of rows from the dataset for building the model.

*Curated dataset: From the disinfo dataset compiled from e.g. FactCheck.org and Wiki, the authors excluded any sites that didn’t align with their definition of disinfo. For inactive sites, the authors reconstructed domain, certificate, and hosting features (some 60% sites – 368 in all – required this). From their local news sub-corpus, the authors removed all websites that didn’t prominently display news; from their overall news corpus, the authors excluded the 100 most well-known news sites, in case these very well-known sources had unique features that distinguished them from the long tail of lesser known news sites.*

*Real-time pilot: The authors manually labeled a random sample of 300 websites extracted by the Twitter API according to the three-category taxonomy described earlier.*

13) How was the train-test split observed during each pre-processing step? If applicable, use a separate line for each step mentioned in Q12.

*Unclear to me how this split was made, so unclear how it was observed during pre-processing.*

14) List all the modeling steps used in creating your model. For example, feature selection, parameter tuning, model selection.

*Feature selection – the authors engineer a set of 33 infrastructure features*

*Model selection – the authors chose a multiclass random forest model because they ‘expected that feature interactions would contribute to performance and that model interpretability would be important for evaluation and plausible deployment’*

*Parameter tuning – the authors conducted a randomized hyperparameter search over a range of values (unspecified), then selected optimal hyperparameters based on classifier performance with five-fold cross-validation.*

*Additionally, the authors trained three standalone models for three subgroups of features within the disinfo category only. These three subgroups were domain, certificate, and hosting feature categories. Domain features ‘predominantly drove classifier performance.’*

15) How was the train-test split observed during each modeling step? If applicable, use a separate line for each step mentioned in Q14.

*Unclear to me how test-train split was made?*

16) List all the evaluation steps used in evaluating model performance. For example, cross-validation, out-of-sample testing.

*Five-fold cross validation to evaluate model performance on the curated dataset.*

*The real-time deployment pilot might be considered an instance of out-of-sample testing?*

*ROC curves to demonstrate performance of model on curated dataset, heatmap to demonstrate performance on real-time pilot.*

17) How was the train-test split observed during each evaluation step? If applicable, use a separate line for each step mentioned in Q16.

*Unclear to me how split was determined.*

**Section 4: Test set is drawn from the distribution of scientific interest.**

18) Why is your test set representative of the population or distribution about which you are making your scientific claims?

*To the extent that I understand the distinction between the test and train set, there are certain biases (some of which the authors actually mention in-text):*

* *34% of websites in the misinfo training data are active, while* all *websites in the test set are active*
* *the curated dataset for disinfo sites was drawn from authoritative sources, including FactCheck.org and Snopes, which might have a bias towards well-known + older disinfo sites*

19) Explain the process for selecting the test set and why this does not introduce selection bias in the learning process.

*The disinfo test set was sampled from sets compiled by well-known news and fact-checking sites, so are more likely more well-known than randomly-sampled sites.*

20) In case your model is used to predict a future outcome of interest using past data, detail how data in the training set is always from a date earlier than the data in the test set.

*There might be some temporal leakage in the curated website set, as the authors state that this dataset is a mix of current and historical data, and it’s unclear to me how any time differences are accounted for during test-train split.*

*sayash’s defn of temporal leakage: withheld set would have to be prior to trained-on set. [predicting things from the past w data from future] [training on things from past, testing on things from current / future – correct protocol]*

**Section 5:** **Each feature used in the model is legitimate for the task**

21) List the features used in the model, alongside an argument for their legitimacy. A legitimate feature is one that would be available when the model is used in the real world and is not a proxy of the outcome being predicted. You can also include this list in an appendix and reference the relevant section of your Appendix here.

*A screenshot of the full feature list is on the next page. The authors determined their feature list by examining infrastructure features of known misinfo and news sites. They were particularly interested in features that are present at the very beginning of a website’s life.*



train: on full dataset – all 1500 websites

test: on twitter API – some 380 websites

validation: on disinfo – on full dataset, not twitter