Problem statement – Fair treatment by a BERT-based Twitter disinformation classifier

We use a quantitative bias scan tool to assess fair treatment of a self-trained disinformation detection algorithm on Twitter data. This document presents statistically significant disparities found by the tool. The results are submitted to a commission of human experts. This audit commission formulates normative advice if, and how, (multi-dimensional) proxy discrimination and/or ethically undesirable forms of differentiation could be investigated further.

1. Introduction

Unfair treatment by algorithms is multi-faceted. A first concern is one-dimensional proxy discrimination. Proxy discrimination concerns unlawful differentiation based on an apparently neutral feature (such as *literacy rate*) that is critically linked to a protected ground as specified in legal directives[1] (such as *ethnicity*). A second concern is ethically undesirable forms of differentiation. Algorithms can differentiate upon a seemingly innocuous feature, such as browser type or house number suffix. This type of differentiation evades non-discrimination law, as many features are not critically linked to a protected ground, but can still be perceived as unfair, for instance if it reinforces social-economic inequality. A third concern is higher-dimensional forms of unfair treatment. Algorithms differentiate upon clusters that are defined by a mixture of features. Higher-dimensional forms of algorithmic differentiation are difficult to detect for humans. Let alone to assess whether the cluster is involved in proxy discrimination and/or ethically undesirable forms of differentiation. In theory, statistical methods are capable to detect both higher- and one-dimensional forms of undesirable differentiation. In this case study, we use a statistical bias scan tool to examine in practice whether the above concerns can be overcome.

2. Unsupervised bias scan

The bias scan tool[2] identifies clusters for which a binary classification algorithm is systematically misclassifying, i.e., predicting a different class than the ground truth label in the data. A cluster is a group of datapoints sharing similar features. The tool makes use of unsupervised clustering[3] and therefore does not require *a priori*

[1] In the European Union (EU), the European Convention of Human Rights (ECHR) serves as the legal fundament against discrimination. Additional EU directives (2000/43/EC, 2000/78/EC, 2004/113/EC, and 2006/54/EC) provide context-specific protection, e.g., persons with disabilities, labor law, and good and services.

[2] Misztal-Radecka, Indurkya, Bias-Aware Hierarchical Clustering for detecting the discriminated groups of users in recommendation systems, *Information Processing and Management* (2021).

[3] Documentation about the k-means Hierarchical Bias-Aware Clustering (HBAC) algorithm can be found here: https://github.com/NGO-Algorithm-Audit/Bias_scan/blob/master/Bias_scan_tool_report.pdf

information about existing disparities and protected attributes of users (which are often not available in practice).

For this case study, we review a BERT disinformation classification algorithm[4] which is trained on the Twitter1516 dataset[5], enriched with self-collected Twitter API data[6]. The dataset consists of 1,057 verified true and false tweets, 3 user features (verified profile, #followers, user engagement) and 5 content features (length, #URLs, #mentions, #hashtags, sentiment score). We run two bias scans. In Scan 1, the bias metric is defined by the False Positive Rate (FPR). In Scan 2, the bias metric is defined by the False Negative Rate (FNR). FPR relates to true content predicted to be false, proportional to all true content. FNR relates to false content predicted to be true, proportional to all false content. In sum:

Scan 1. $\text{Bias} = \text{FPR}_{\text{cluster}} - \text{FPR}_{\text{rest of dataset}}$
Scan 2. $\text{Bias} = \text{FNR}_{\text{cluster}} - \text{FNR}_{\text{rest of dataset}}$.

The full bias scan pipeline is displayed in Figure 1.

Figure 1 – Bias scan pipeline for the disinformation classifier.

3. Results: Identified quantitative disparities

For Scan 1, the cluster for which the disinformation classifier is underperforming the most (bias=0.08, n=249) is characterized by the features displayed in Figure 2. The feature difference is the average in means between the disparately treated cluster and the rest of the dataset. Hypothesis testing[7] indicates that on average, user that:

- are verified, have higher #followers, user engagement and #URLs;
- use less #hashtags and have lower tweet length

have more true content classified as false (false positives).

---

[4] More information about the self-trained BERT-based classification algorithm can be found here: https://github.com/NGO-Algorithm-Audit/Bias_scan/blob/master/case_studies/BERT_disinformation_classifier

[5] Liu, Xiaomo and Nourbakhsh, Armineh and Li, Quanzhi and Fang, Rui and Shah, Sameena, in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (2015)

[6] More information on the data collection process can be found here: https://github.com/NGO-Algorithm-Audit/Bias_scan/blob/master/data/Twitter_dataset/Twitter_API_data_collection.ipynb

[7] Here, the hypothesis tested is that there is no difference in feature means of the cluster and the pooled feature means of other clusters. These differences are statistically significant even after performing a Bonferroni correction to adjust for false discoveries due to multiple hypothesis testing.

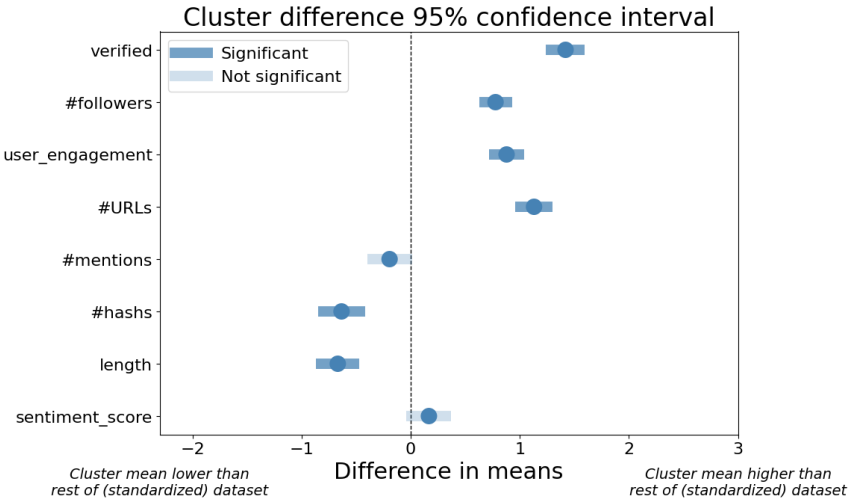

Figure 2 – Identified quantitative feature disparities in cluster with highest bias. For this bias scan, bias is defined by the False Positive Rate.

For Scan 2, the cluster for which the disinformation classifier is underperforming the most (bias=0.13, n=46) is characterized by the features displayed in Figure 3. Hypothesis testing indicates that on average, user that:

- use more #hashtags and have higher sentiment score;
- are non-verified, have less #followers, user engagement and tweet length have more false content classified as true (false negatives).

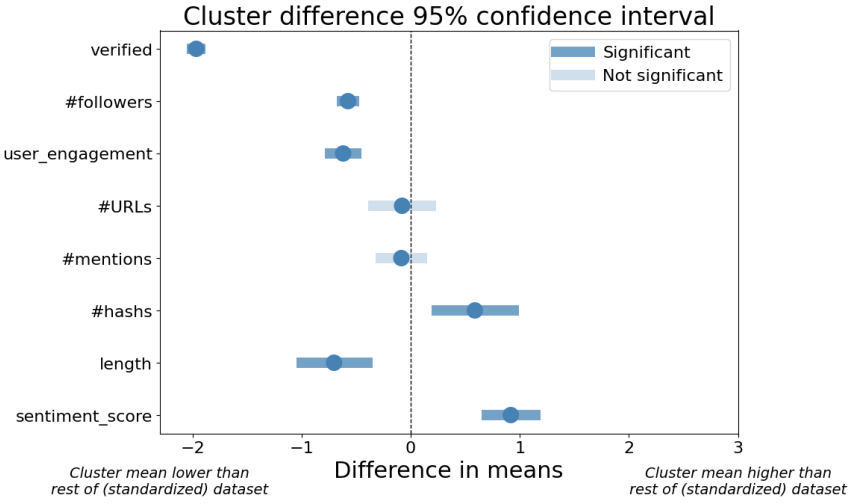

Figure 3 – Identified quantitative feature disparities in cluster with highest bias. For this bias scan, bias is defined by the False Negative Rate.

These results might indicate (higher-dimensional) unfair treatment by the disinformation classifier. More information on the identified clusters and robustness tests of the results can be found in the Appendix.

4. Qualitative assessment of identified disparities

The identified disparities in Section 3 do not establish prohibited *prima facie* discrimination. Rather, the identified disparities serve as a starting point to assess potential unfair treatment according to the context-sensitive qualitative doctrine. To assess unfair treatment, we question:

i) Is there an indication that one of the statistically significant features, or a combination of the features, stated in Figure 2-3 are critically linked to one or multiple protected grounds?

ii) In the context of disinformation detection, is it as harmful to classify true content as false (false positive) as false content as true (false negative)?

iii) For a specific cluster of people, is it justifiable to have true content classified as false 8 percentage points more often? For a specific cluster of people, is it justifiable to have false content classified as true 13 percentage points more often?

iv) Is it justifiable that the disinformation classification algorithm is too harsh towards users with verified profile, more #followers and higher user engagement and too lenient towards users with non-verified profile, less #followers and lower user engagement?

Auditing disinformation detection algorithms

As of December 2022, Article 28 of the European Digital Services Act (DSA) subjects very large online platforms (VLOPs) to annual independent auditing of their services and risk mitigation measures. Open-source AI auditing tools, such as this bias scan tool, help to detect and mitigate (higher-dimensional) forms of unfair treatment in disinformation detection and other AI (ranking or recommender) systems.

With this case study, Algorithm Audit aims to provide qualitative guidelines how statistical methods can be used to monitor unfair treatment by AI systems. Without clear guidance from data protection boards, supervisors, researchers and algorithmic regulatory body, misinterpretation of quantitative metrics stands in the way of independent quantitative and qualitative oversight of the risk of biased AI systems. Algorithm Audit provides normative advice by convening audit commissions to shed light on identified ethical issues.

Appendix

BERT-based classifier performance

The confusion matrix of the BERT-based disinformation classifier is displayed in Figure 4. More information regarding the training process can be found on Github[4].

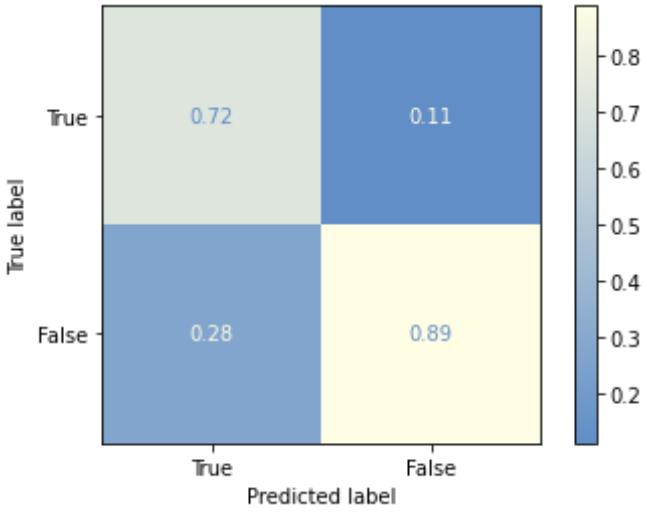

Figure 4 – Confusion matrix of the BERT-based disinformation classifier on the test set.

Bias scan results

For the FPR scan, the following clusters are detected:

- Cluster 0 has bias -0.062
- Cluster 1 has bias 0.0810
- Cluster 2 has bias -0.089
- Cluster 3 has bias -0.041

Cluster 1 has the highest bias (FPR): 0.08. There are 249 elements in cluster. Table 1 displays all numeric values of the difference in feature means between the cluster and the rest of the dataset, including p-values of a Welch's two-samples t-test for unequal variances to examine whether the differences are statistically significant (p-value<0.05). Note that Table 1 is displayed in Figure 2.

	Difference	p-value
Verified profile	1.419	0.000
#followers	0.778	0.000
User engagement[8]	0.878	0.000

[8] For user engagement metric see: https://developer.twitter.com/en/docs/twitter-api/enterprise/engagement-api/overview

#URLs	1.130	0.000
#mentions	-0.669	0.064
#hashtags	-0.634	0.000
Length	-0.669	0.000
Sentiment score[9]	0.167	0.115

Table 1 – Difference in feature means between cluster with highest bias (FPR) and the rest of the dataset. Rows in blue display a statistically significant difference according to a Welch's two-samples t-test for unequal variances (p<0.05).

For the FNR scan, the following clusters are detected:

- Cluster 0 has bias -0.101
- Cluster 1 has bias 0.118
- Cluster 2 has bias -0.059
- Cluster 3 has bias 0.132

Cluster 3 has the highest bias (FNR): 0.13. There are 46 elements in cluster. Table 2 displays all numeric values of the difference in feature means between the cluster and the rest of the dataset, including p-values of a Welch's two-samples t-test for unequal variances to examine whether the differences are statistically significant (p-value<0.05). Note that Table 2 is displayed in Figure 3Figure 2.

	Difference	p-value
Verified profile	-1.965	0.000
#followers	-0.575	0.000
User engagement[8]	-0.619	0.000
#URLs	-0.079	0.607
#mentions	-0.086	0.465
#hashtags	0.588	0.004
Length	-0.702	0.000
Sentiment score[9]	0.917	0.000

Table 2 – Difference in feature means between cluster with highest bias (FNR) and the rest of the dataset. Rows in blue display a statistically significant difference according to a Welch's two-samples t-test for unequal variances (p<0.05).

[9] For sentiment score see: https://github.com/cjhutto/vaderSentiment.

Sensitivity testing

The k-means HBAC algorithm uses various hyperparameters. In this section, we provide a rationale for our choices for these parameters. In addition, we refer to sensitivity testing that confirms the results as presented in Section 3.

Parameters prevent HBAC to find only clusters with a small amount of datapoints, for which it is hard to find meaningful features. An overview and description of all hyperparameters is given in Table 3.

Number of initial clusters (Our choice: 2)	The desired number of initial clusters oft he k-means clustering algorithm.
Maximum number of iterations (Our choice: 300)	The HBAC algorithm is terminated after the maximum number of iteration threshold is reached, or after no clusters are found that have a higher discrimination bias when compared to the clusters of the previous iteration.
Minimal splitable cluster size (Our choice: 29)	Number of elements that need to be in the cluster to be eligible for a next cluster split.
Minimal acceptable cluster size (Our choice: 21)	Number of elements in a new candidate cluster during splitting to be accepted as a new cluster.

Table 3 – Hyperparameters of the HBAC algorithm.

We run the FPR and FNR scan for the following 162 configuration of hyperparameters:

- Number of initial clusters: 2 and 3;
- Minimal splitable cluster size: 5, 10, 15, 20, 25, 30, 35, 40 and 45;
- Minimal acceptable cluster size: 5, 10, 15, 20, 25, 30, 35, 40 and 45.

We compute the average of all cluster with positive (FPR or FNR) bias. This results in:

- 2974 clusters with positive FPR bias;
- 2506 clusters with positive FNR bias.

More information on the results of these sensitivity tests can be found on GitHub[10].

[10] https://github.com/NGO-Algorithm-Audit/Bias_scan/blob/master/HBAC_scan/HBAC_BERT_disinformation_sensitivity_testing.ipynb