Probing the third-party infrastructure of digital news on the Web

Ido Sivan-Sevilla
The University of Maryland
College Park, Maryland, USA
sevilla@umd.edu

Parthav Poudel
The University of Maryland
College Park, Maryland, USA
ppoudel1@umd.edu

Abstract

The wide spread of disinformation across news websites makes fast, efficient, and reliable detection of untrustworthy content more important than ever. Conventional methods for detecting fraudulent content on the Web rely on content- or social network-based analysis. In contrast, we build on previous work to further explore whether the features and attributes of the third-party request structures of websites can be used at scale to distinguish between fake and real news content on the Web. We crawled 5,478 real and fake news websites that are already labeled by NewsGuard, on a daily basis, over the course of seven months, and collected data on their changing third-party structure, extracting static and temporal structural features. We show promising accuracy results for our Random Forest prediction model, solely based on structural features. We also reveal several key indicators in websites' structural trees, including (1) higher 7-day average of resource requests per node, and (2) a greater maximum breadth of resource request trees, that are likely to indicate trustworthy content. Our method can be used to complement current content- and social-network-related prediction methods when they are indecisive about fake news content on the Web.

Keywords

Digital News, Third-party Structure, Trustworthy Website Measurement

1 Introduction

The spread of false or misleading information across websites calls for methodologies that can recognize, predict, and flag fake content. Existing methodologies are typically based on analyzing content or use the social network settings that evolve around content spread, with scholars recently applying text-based algorithms (including LLMs) to assess whether online content is reliable [1, 8–10, 13]. The exclusive focus on content classification or settings around content distribution makes the detection and enforcement of policies against misinformation a lengthy task. The complexities of content analysis might challenge real-time detection of fake content, before it spreads and mobilizes individuals, institutions, and nations. LLMs have made the task easier, but they still surface difficulties in quantitative and qualitative content analysis and introduce costs and latency in the recognition process. Structural features of the hosting websites, however, are often overlooked and can serve as a useful proxy for the reliability of online content. Structural characteristics indicate which resources websites use, what the funding structure of a website is, and how these attributes change over time.

We follow and further expand existing work to explore to what extent the 'behind-the-scenes' third-party structure and features of websites can be used to predict the level of trust in site's content [5]. Our research motivation comes from the heavy reliance of any website, trustworthy or not, on a variety of resources that are either hosted locally or by third parties. These can include third-party services that contribute to functionality, performance, user analytics, security, and advertising purposes. Outsourcing website operations is expressed by requests sent by the website to an array of third-parties, or by the third-party themselves to other third parties. These requests create a third-party request structure that can serve as a proxy to understand how a website is maintained, functions, and monetizes its content. We seek to explore whether this third-party structure can also serve as a proxy for the level of trust in site's content.

The ability to generate detection methodologies that work on a scale and are agnostic to website content is the main contribution of this paper. To meet this challenge, we crawled 5,478 real and fake news websites, based on their labeling by NewsGuard, and collected data on their third-party request structure. We collected the data on a daily basis, from February 17, 2024 to September 19, 2024, and assembled 216 third-party daily trees (at most) for each crawled website. We used these observations to train a supervised machine learning algorithm to classify resource request trees of websites from real vs. fake news websites based on their structural features.

2 Can Structure Predict Content?

The detection of fraudulent content on the Web usually evolves around content-based features and social media contexts. Content-based detection is based on textual or linguistic features. Social context-based features include user behavior and social media patterns around the spread of content [1–3, 6, 10, 13]. Existing detection methods often evaluate and identify the trustworthiness of websites at the article level, rather than identifying real vs. fake news websites as a whole.

Han et al. (2022) & Gopel et al. (2022) diverge from these detection trends. Han et al. (2022) inspected the service providers of misinformation websites and found that several providers are disproportionately responsible for serving misinformation websites, and that misinformation sites disproportionately rely on several popular ad networks and payment processors, including RevContent, PayPal, and Google DoubleClick [7]. The work provides a motivating starting point to further inspect the resources and outsourcing operations of misinformation sites, but the findings are not utilized for prediction purposes.

Free and Open Communications on the Internet 2025(2), 74–81

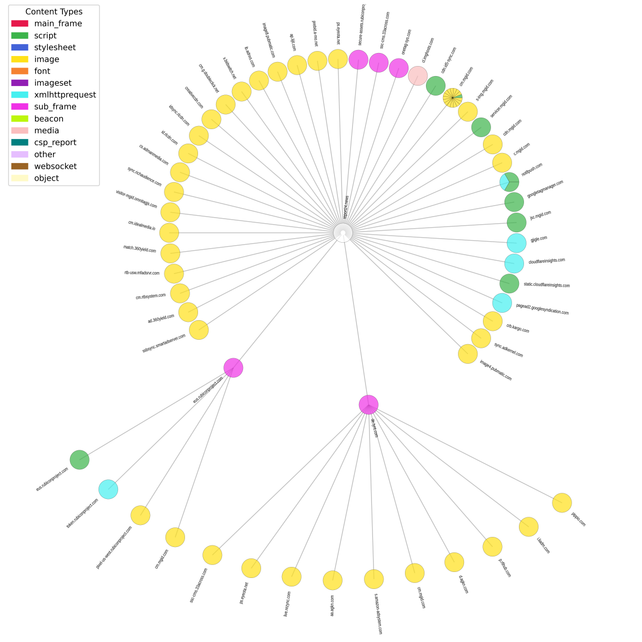

Figure 1: Example of a third-party structure of a fake news website: report24.news

Gopel et al. (2022) took it a step further. They use the third-party request structures of websites to detect fraudulent websites, across an array of phishing, fake e-commerce, fake news, and piracy websites [5]. Their results are promising, but are only based on 205 websites, 32 of them in the news category. We follow their call to increase sample size and focus specifically on mis- and dis-information news sites to realize whether the structural features of a website can predict the level of trust in its content.

The resource request structure of a website is determined by calls to various first- and third-party resources. The direct calls from a website to its third parties are embedded in the HTML code of the website. The third-party structure also includes indirect calls, from third-parties to other third-parties, which are not embedded in the HTML code of the website itself but can be identified in runtime.

The third-party structure of a website is visualized in figure 1. Each node represents a third-party receiving (or sending) a request. Edges represent requests. Each request asks for a specific type of service, and each third-party can appear multiple times. Third parties that appear in the first level of the tree are directly requested from the website. Any level beyond that indicates that a third-party was requested from another third-party rather than the website itself. Figure 1 visualizes a third-party structure example of report24.news, a news site that is labeled as 'non-trustworthy' by NewsGuard. The colors for each node indicate the content type provided by the first- or third-party resource. Some third parties can offer various content types and are multi-colored accordingly.

Interestingly, third-party structures of websites tend to change over time. Figure 2 shows those changes in the third-party tree features of the report24.news website. We see how the total number of requests to resources (first and third parties) by the website, marked

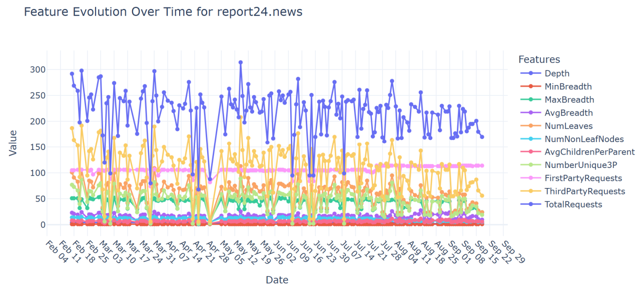

Figure 2: Feature Evolution Chart for report24.news

with the blue color, rapidly changes over time. Those dynamics in third-party structures per website motivated us to explore temporal, not only static dimensions of websites' resource reqest trees.

3 Methodology

To analyze the resource request structures of real and fake news websites and their temporal variations, we conducted a large-scale data collection effort spanning seven months. Between February 17, 2024 and September 19, 2024, we continuously monitored and collected valid third-party request data from 5,478 unique domains on a daily basis. During the data collection period, some websites went offline or failed to be crawled consistently. To ensure reliability, we filtered out domains with fewer than 50 successful crawl days. As a result, 5,123 domains were crawled for at least 50 days, 2,180 domains for at least 100 days, and 1,261 domains for 150 days or more. These subsets formed the basis of our analysis and model evaluation, enabling us to study structural behavior across short-, medium-, and long-term time windows.

The domains under study were sourced from NewsGuard, a professional media reliability assessment service, and included both trustworthy and untrustworthy news outlets. The commercial NewsGuard service deploys the "News Reliability Ratings" that is based on a team of expert journalists that rate and review the reliability of news sites across the open Web. Sites are assigned ratings based on a score derived from nine journalistic criteria assessing credibility and transparency. For our study, we focused specifically on the "T" rating (Generally Trustworthy, score of 60 or above) and the "N" rating (Generally Not Trustworthy, score below 60). To train our model, we did not include sites under NewsGuard's "FL" rating (Flagged as having serious trust issues and pending review by NewsGuard). Sites categorized as Satire, Platform, or other special designations were also excluded from our analysis to ensure clarity in the classification of trustworthy versus untrustworthy content. Newsguard's Reliability Ratings became a standard among researchers and practitioners for detecting mis- and dis-information sites. The actual methodology behind the ratings, however, is unclear, and the time lag between the availability of a news website and its detection as fraudulent is uncertain. That time lag can be crucial as fake news tends to spread quickly and can rapidly cause damage.

To ensure consistency and reliability across our data set, we only used websites for which we could collect at least 50 days of data during the 216 days period. Some websites became offline during

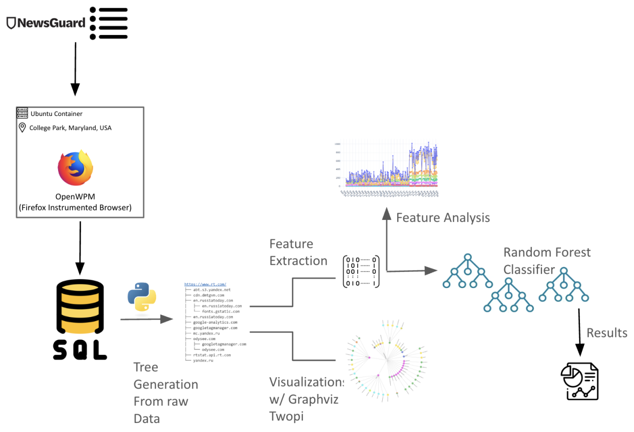

Figure 3: Data Collection & Analysis Diagram

our crawl period, while others were not consistently crawled for unknown reasons.

As a result, our dataset contained a class imbalance, with significantly more trustworthy (n=3645) than untrustworthy (n=1478) news websites that have at least 50 data points. To mitigate selection bias, we balanced the training data by up-sampling the minority class (untrustworthy sites) to fit the majority class (trustworthy sites) size during model training. We did that by adding up to four observations from different dates on the same untrustworthy site.

The following subsections detail the various data collection, analysis, and classification tasks that were preformed. We assembled a (1) crawling infrastructure from which we derived the (2) construction of resource request trees for each website, and (3) extracted the temporal features of each tree attribute to trace how resource requests change over time. Our collected data served as input to (4) our chosen supervised classification model - the Random Forest classifier - to detect and predict fake and real news websites based on structural features. Figure 3 illustrates our data collection and analysis processes.

3.1 Crawling Infrastructure

To ensure robust and repeatable data capture, we deployed `OpenWPM` [4], a Web privacy measurement platform, on a dedicated server located at the University of Maryland, College Park. The server was equipped with 1TB of RAM to support concurrent and memory-intensive crawling operations. We launched crawls against every domain on our list, capturing HTTP request logs, JavaScript activity, and first- and third-party resource loads. The crawling pipeline resulted in more than 3TB of raw data, encapsulating detailed structure dynamics for each domain over time. Importantly, we excluded 152 domains that made bot detection-related calls (e.g., requests to `captcha-delivery.com` or `perimeterx`) to minimize the influence of anti-crawling mechanisms on our data.

3.2 Third-Party Tree Construction

Following data collection, we reconstructed third-party request structures (trees) for each [website,date] pair from the raw crawl data. In each tree, nodes represent distinct third-party domains, and edges denote the hierarchical sequence of HTTP requests. These features were chosen because they collectively represent the complexity, scale, and variability of third-party dependencies utilized by websites.

The dependent variable of the study is a binary classification of *UntrustworthyNewsWebsite* which is a 0,1 binary categorical variable (0 if Trustworthy, 1 if Untrustworthy). Independent variables for the prediction model include the *Depth* of the structure tree, *MinBreadth* at any tree level, *MaxBreadth* at any tree level, *AvgBreadth* across all tree levels, *NumLeaves*, *NumNonLeafNodes*, *AvgChildrenPerParent*, *NumUnique3P*, which is the number of unique third-parties per tree, *FirstPartyRequests*, which is the number of times that a website seeks local resources, *ThirdPartyRequests*, which is the number of times a website seeks external resources, and *TotalRequests*, which summarizes the entire call to any resource by the website.

Descriptive statistics for each extracted feature, across trustworthy and untrsutworthy sites, as well as the description of each feature, are presented in Table 1.

3.3 Temporal Feature Engineering

To capture the temporal changes in the resource request structures of websites, we calculated a variety of temporal features for each of the structural attributes described in Table 1. These features are detailed Table 2. The temporal features capture how consistent or volatile a website third-party structure is, between days of observation over the course of seven months. They include vectors of: rolling 7-day means of each structural attribute, rolling 7-day standard deviation of each structural attribute, day-to-day differences in each structural attribute, and day-to-day frequency of change in each structural attribute. The multi-layered temporal modeling provided insights into both short-term fluctuations and long-term behavioral trends in the way websites handle their resource request.

3.4 Classification Task and Model Details

We treat our website classification as a binary classification task to predict whether a news website is *trustworthy* or *untrustworthy*, solely based on the static and temporal structural characteristics of its resource request trees. The ground-truth label (`Fradulent`) is derived from NewsGuard trust scores, as described earlier.

We train three Random Forest models, depending on how many days a domain was observed (>50, >100, >150). The input matrix per domain included 55 columns - one column for each of the 11 input variables that were measured daily, and four additional columns per variable as detailed below:

- **Rolling means and SDs (7-day)**: Rolling Averages and Standard Deviations of the 11 structural features to capture short-term trends.
- **Daily deltas of change**:The change in the structural feature value between days of measurement.
- **Daily frequency of change**: Has the structural feature changed between two days of measurement (0 or 1)?

To balance the training data between trustworthy and untrustworthy domains, we used the following as input to the Random Forest Classifer: one row from the input matrix for any trustworthy domain and up to 4 rows from the input matrix of the same

		Trustworthy				Untrustworthy			
Feature	**Description**	Mean	Median	Min	Max	Mean	Median	Min	Max
Depth	Maximum depth of the request tree	2.32	2.0	1.0	15.0	2.13	2.0	1.0	6.0
MinBreadth	Minimum breadth at any level	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
MaxBreadth	Maximum breadth at any level	39.22	26.0	1.0	539.0	19.63	12.0	1.0	279.0
AvgBreadth	Average breadth across all levels	16.19	10.67	1.0	206.0	8.6	5.67	1.0	145.4
NumLeaves	Count of terminal nodes	52.19	30.0	1.0	1177.0	25.58	14.0	1.0	668.0
NumNonLeafNodes	Non-leaf node count	7.03	5.0	0.0	86.0	5.42	4.0	0.0	149.0
AvgChildrenPerParent	Avg children per non-leaf	8.35	7.0	0.0	108.0	4.79	4.0	0.0	88.0
NumUnique3P	Unique third-party domains	44.15	25.0	0.0	414.0	20.31	11.0	0.0	316.0
FirstPartyRequests	Originating domain requests	44.42	34.0	0.0	3497.0	47.34	35.0	0.0	2971.0
ThirdPartyRequests	External domain requests	137.45	79.0	0.0	10203.0	74.12	33.0	0.0	3944.0
TotalRequests	Sum of all requests	181.87	131.0	0.0	10243.0	121.46	86.0	0.0	3955.0

Table 1: Extracted structural features and their statistics from third-party request trees across all crawled domains

untrustworthy site, making the site appear up to four times in our training data, from different dates of observation.

We use a domain-level stratified train/test split (70% training, 30% testing) to ensure that all records for a given domain appear exclusively in either the training or testing set.

Following these model decisions, we trained three different Random Forest models, for domains with (1) at least 50; (2) at least 100; and (3) at least 150 days of observation. We had 4,660 unique domains for the first group, 2,100 domains for the second group, and 961 domains for the third group.

We train a `RandomForestClassifier` from the `scikit-learn` Python library with the following parameters:

- `n_estimators=200`: Specifies the number of trees in the forest.
- `class_weight='balanced'`: To handle the class imbalance as our data contains more untrustworthy sites than trustworthy ones.
- `max_depth=None`: Allows trees to grow until all leaves are pure or contain fewer samples than the minimum split threshold. This enables the model to fully fit the data unless explicitly restricted.
- `random_state=42`: Ensures reproducibility by controlling the randomness of bootstrapping and feature selection in each tree.
- `n_jobs=-1`: Enables parallel computation.

While default values for `n_estimators` like 100 are commonly used, increasing to 200 often stabilizes performance and reduces variance in prediction without significant additional cost [12], especially with parallel processing enabled.

Temporal Feature	Description
Rolling Mean (7-day) [vector]	The 7-day rolling average of each structural feature for a domain. For instance, if a site's number of unique third-party domains contacted changes daily, this value provides a 7-day window of observation on that value change.
Rolling Standard Deviation (7-day) [vector]	The 7-day rolling standard deviation of a feature, indicating short-term volatility. A consistently structured site will have low standard deviation, whereas a site that frequently adds/removes third-party calls will show higher variation.
Daily Delta [vector]	The day-to-day difference in a given feature's value. For example, if the number of first-party requests jumps from 80 to 100 between two days, the delta is +20.
Delta Frequency [vector]	Counts the frequency of changes in a structural feature between days of observation. For example, if the third-party tree depth changed on 90 out of 150 days, the delta frequency will be +1 in each of those days. This captures how "active" or "stable" a site's structure is.

Table 2: Engineered temporal features used to capture structural consistency, volatility, and change across time

In future work, we plan to explore temporal classifiers such as Time Series Forests to better model sequential dynamics, particularly for domains with many daily observations.

4 Limitations

Limitations to our methodology include the reliance on a single classifier - the Random Forest Classifier - to predict and classify fake content based on structure. We plan to examine additional models, including time-series models, and compare the results.

Additional limitations stem from the detection of our crawler by websites which might cause selection-bias, and disproportinally reduce the number of untrustworthy sites we can train our model on. Websites on our list deployed anti-bot measures which prevented us from crawling these sites. Specifically, 152 domains presented captcha calls and blocked our automatic crawling. Looking ahead, we plan to follow best practices from the literature and implement various improvements to OpenWPM crawling to prevent potential bot detection [11]. To mitigate un-balanced training data at the moment, we used up to four observations of the same untrustworthy sites from different dates.

Another limitation is the lack of categorization of the third-party domains found in the data. We know from previous research that third-parties are embedded in websites for an array of advertising, analytics, security, and functionality reasons. We plan to properly classify the array of third-parties found in our data to understand which types of third-parties are more common on fake vs. real news websites and for which purposes.

5 Results

The descriptive statistics in Table 1 help illustrate general patterns and variations within our dataset between trustworthy and untrustworthy sites, providing context for the predictive modeling that was performed. We can see clear differences between trustworthy and untrustworthy sites in the breadth, average calls from each first or third party resource of a website (avg children per parent), number of unique third parties that websites use, and number of requests to third parties overall. Trustworthy sites use much more external resources and call much more third-party assets during their loading than untrustworthy sites. This is in sharp contrast to the findings from Han et al. (2022) [7]. The intuition for this finding is that trustworthy sites are more focused on user experience and tend to be more complex in their resource usage, while untrustworthy sites tend to be more simple, probably focusing on the spread of the content rather than the quality of the user experience.

The results of our Random Forest Classifier show that we can use these structural differences to predict how reliable the content of a news website is. In the subsections below, we detail the quantitative analysis of sites' structural features and provide qualitative insights on the type of resources used by news websites.

5.1 Quantitative Analysis of Digital News Trees

To predict whether a domain is trustworthy, we trained Random Forest classifiers using structural features of third-party request trees, incorporating both static and temporal variables as detailed in Section 3.4. To ensure data reliability and consistency, we filtered domains based on the number of successful crawl days. Specifically, we analyzed results for domains with at least 50 (5,123 domains), 100 (2,180 domains), and 150 (1,261 domains) days of observation.

These features capture both short-term and long-term dynamics in a site's third-party structure. Including both the delta in the frequency of structural change and the delta of the magnitude of change. This had allowed us to distinguish between websites with consistently stable behavior versus those that frequently *and* drastically changed their third-party structure. To train our three models based on balanced training data set, we included one observation on each trustworthy site, and up to four observations, from different dates, on untrustworthy sites from our sample (as explained in Subsection 3.4). The confusion matrix for each of the three models appears in the appendix and their performance results are below.

As shown in Table 3, our model performs best when more domains are used to train the model. For domains with at least 50 observations, we achieve the best results. We had 4,660 unique domains and balanced our training data with observations of the same untrustworthy site from different dates. The model reports ROC AUC of 0.81 and balanced precision/recall (0.72 for both categories on average). This means that structural features are useful for correctly distinguish between fake and trustworthy news websites. As we reduce the number of domains used to train the model, by requiring at least 100 or 150 daily observations per website, performance drops slightly. The two additional models report ROC AUC of 0.78 & 0.68 and precision average of 0.71 or 0.65 on average for both categories. Average recall in both models, for both categories drop as well, with 0.71 and 0.65 average recall for both categories in the two models.

The slightly lower recall numbers in Table 3 in domains with more daily observations can be mitigated once we will be able to scan more sites that were classified as untrustworthy, successfully crawling the domains that have been blocking us. With more training data on untrustworthy domains, we expect the slightly lower recall numbers to improve.

To verify that the number of domains in the training data has an impact on model performance, we trained additional Random Forest Classifiers based on randomly selected domains from the set of 4,660 domains, with at least 50 observations. We selected 1,280 and 2,100 domains from that sample to create two additional models and compare them with the results in Table 3. We witnessed a similar decline in model performance when fewer domains were used to train the model.

Figure 4 shows the importance of extracted features from resource request trees for the Random Forest model that is based on domains with at least 50 days of observation. The most importnat features contributing to the model's predictive performance are: (1) the average number of children per non-leaf node in the resource request tree; (2) the 7-day rolling average of maximum breadth of the tree; (3) the 7-day rolling average of average breadth of the tree; (4) the average number of childs per parent in the three; and (5) the 7-day rolling average of third-party requests by the website.

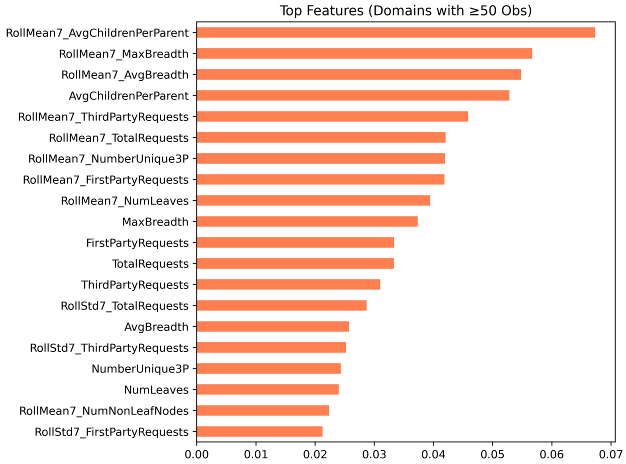

Figure 4: Top 20 most important features when training the model on domains with at least 50 days of observations

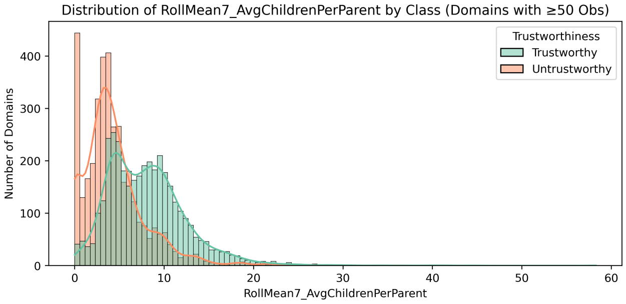

Figure 5: Distribution of 7-day avg of children per tree node across domains with at least 50 days of observation

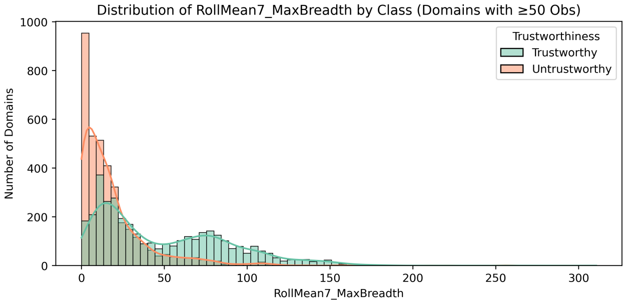

Figure 6: Distribution of 7-day rolling average of maximum breadth of the resource request tree across trustworthy and untrustworthy news domains

Figure 5 shows the impact of the average number of children per node on the distribution of fake and real news domains. The more calls to third-party resources by a tree node (the website itself or other third party), the higher the likelihood that the tree represents a trustworthy news website. This is in contrast to the findings of Han et al. (2022), who found that misinformation sites are more heavily relying on third parties [7]. Our intuition for these results is that untrustworthy sites are under-resourced and built in a rather straightforward way to make the content spread rather than to provide a cutting-edge browsing experience for their users.

Figure 6 shows the impact of the average maximum breadth of a tree on the classification of website content. The higher the maximum breadth per tree is, the higher the likelihood that the tree represents a real news website. The intuition for the results is that trustworthy websites keep improving their operations and updating their calls to resources, while untrustworthy sites tend to stick to the minimal working arrangement for their operation. These sites are designed with very different goals in mind: trustworthy sites pay great attention to user experience, while untrustworthy ones mostly focus on wide spread of content and tend to keep operation and resource request simple.

5.2 Qualitative Analysis of Digital News Trees

Following the promising prediction trends observed in the quantitative analysis, we dived deeper to assess the type of third-parties that are involved in fake vs. real news websites. We assembled a list of all third parties seen for each domain, and categorized it to trustworthy vs. untrustworthy websites from which the calls were made. For each third-party we counted the frequency of its appearance in fake vs. real news websites.

We created a list of third parties that appear on 3,410 websites, 1,705 from each category, based on NewsGuard labeling. In our initial data collection we ended up with 3,773 real news websites

Min Daily Obs	# T	# N	Domains	Rows	Acc.	Prec. (T)	Rec. (T)	F1 (T)	Prec. (N)	Rec. (N)	F1 (N)	ROC AUC
≥ 50	3 684	976	4 660	7 368	0.72	0.70	0.77	0.73	0.74	0.67	0.70	0.81
≥ 100	1 608	494	2 100	3 216	0.71	0.69	0.75	0.72	0.73	0.67	0.70	0.78
≥ 150	961	320	1 280	1 922	0.65	0.64	0.70	0.67	0.66	0.59	0.63	0.68

Table 3: Classification performance under balanced-sampling

and 1,705 fake news websites according to NewsGuard. We had to randomly select 1,705 out of 3,773 real news websites to even the sample.

After filtering out unique third-party domains that appear only once across our data set, the vast majority of them are google-syndication domains, we were left with 8,004 third-parties overall. 4,015 of them appear on both trustworthy and untrustworthy websites; 997 of them only appear in untrustworthy websites; and 2,992 exclusively appear in trustworthy websites. Interestingly, domains that exclusively appear on fake news websites are not the well-known advertising or analytics domains. Domains that exclusively appear on trustworthy domains, however, are mostly small-medium advertising and analytics actors.

For domains that appear on both types of websites - some mostly appear in trustworthy ones, while others are more common in untrustworthy websites. Domains that mostly appear in trustworthy sites are the usual advertising and analytics giants - Google, Amazon, Rubicon, Criteo, Adobe, Pubmatic, and etc. This is not surprising, as these actors are the main enablers of content monetization in News websites. Domains that mostly appear on untrustworthy sites include actors such as Google, Zamanta, Yandex, and PayPal. These are actors that are common to the operation of many untrustworthy sites, with PayPal used to accept donations and also appear in the findings of Han et al. (2022). These third-parties also exist in trustworthy sites, but appear more frequently on untrustworthy sites. In contrast to Han et al. (2022), we do not find DoubleClick or Revcontent to be disproportionally used by untrustworthy sites. In fact, according to our data, these actors appear more in trustworthy websites.

We plan to extend this analysis by categorizing each third-party domain into its purpose, which can span across advertising, analytics, security, content delivery, etc. We seek to get a better understanding of how different website functionalities are maintained, and whether there is a unique way of maintaining and operating fake vs. real news sites. We also aim to examine which functionalities in a website are overrepresented in trustworthy vs untrustworthy news sites. The effort will expand our understanding of the characteristics of each third-party category and provide more signals for our classification models.

6 Discussion & Conclusion

Inspecting the structural features of the third-party request trees and the publicly available attributes of third parties embedded in websites can be used to effectively distinguish between fake and real news websites. Tested at scale, across more than 4,500 news websites, over the course of seven months, we showed that various structural features of a website are important for classifying and predicting the level of trust in its content.

Since the classificaiton of mis- and dis-information websites has been far from straightforward, our methodology can complement existing content-focused methodologies of discovery, and can help flag websites that are hard to classify solely based on their content. Our approach is less costly or computationally complex, and much less time-consuming than content-based approaches. We aim to complement, not replace, existing content-based detection methodologies, and offer a promising starting point for a more precise classification of news websites. Particularly, our method can help decide on the level of trustworthiness when NewsGuard labels, for instance, are close to 60 or the site is just 'flagged' for review.

In contrast to content-based approaches, that can be tricked by altering the appearance of content on the Web, third-party structures are more difficult to hide or manipulate as they directly impact the functionality of websites. Theoretically, one could obfuscate the structure of their website based on false/null calls to resources. Such approach, however, will affect site performance and can be spotted pretty easily. We believe that most websites will not follow this self-harming pattern, and if they do, it could be easily flagged through simple scripts.

Our model and results can inform regulators who aim to enforce mis- and dis-information policies, help NGOs and commercial services tag fake content, and most importantly, assist in taking down manipulative content.

Acknowledgments

The authors thank FOCI's four anonymous reviewers who provided constructive and helpful feedback to improve the manuscript.

References

[1] Hadeer Ahmed, Issa Traore, and Sherif Saad. 2017. Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, Issa Traore, Isaac Woungang, and Ahmed Awad (Eds.). Springer International Publishing, Cham, 127–138.

[2] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web* (Hyderabad, India) *(WWW '11)*. Association for Computing Machinery, New York, NY, USA, 675–684. https://doi.org/10.1145/1963405.1963500

[3] Yimin Chen, Niall J. Conroy, and Victoria L. Rubin. 2015. Misleading Online Content: Recognizing Clickbait as "False News". In *Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection* (Seattle, Washington, USA) *(WMDD '15)*. Association for Computing Machinery, New York, NY, USA, 15–19. https://doi.org/10.1145/2823465.2823467

[4] Steven Englehardt and Arvind Narayanan. 2016. Online Tracking: A 1-million-site Measurement and Analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 1388–1401. https://doi.org/10.1145/2976749.2978313

[5] Ram D. Gopal, Afrouz Hojati, and Raymond A. Patterson. 2022. Analysis of third-party request structures to detect fraudulent websites. *Decision Support Systems* 154 (2022), 113698. https://doi.org/10.1016/j.dss.2021.113698

[6] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. 2015. TweetCred: Real-Time Credibility Assessment of Content on Twitter. arXiv:1405.5490 [cs.CR] https://arxiv.org/abs/1405.5490

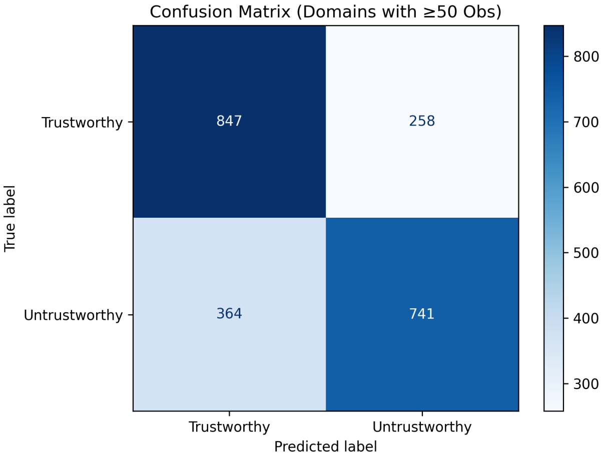

Figure 7: Confusion Matrix for the Model trained on domains with at least 50 days of observation

[7] Catherine Han, Deepak Kumar, and Zakir Durumeric. 2022. On the Infrastructure Providers That Support Misinformation Websites. *Proceedings of the International AAAI Conference on Web and Social Media* 16, 1 (May 2022), 287–298. https://doi.org/10.1609/icwsm.v16i1.19292

[8] Hans W. A. Hanley, Deepak Kumar, and Zakir Durumeric. 2024. Specious Sites: Tracking the Spread and Sway of Spurious News Stories at Scale. In *2024 IEEE Symposium on Security and Privacy (SP)*. 1609–1627. https://doi.org/10.1109/SP54263.2024.00171

[9] Hans W. A. Hanley, Emily Okabe, and Zakir Durumeric. 2025. Tracking the Takes and Trajectories of English-Language News Narratives across Trustworthy and Worrisome Websites. arXiv:2501.09102 [cs.SI] https://arxiv.org/abs/2501.09102

[10] Pakindessama M. Konkobo, Rui Zhang, Siyuan Huang, Toussida T. Minoungou, Jose A. Ouedraogo, and Lin Li. 2020. A Deep Learning Model for Early Detection of Fake News on Social Media. In *2020 7th International Conference on Behavioural and Social Computing (BESC)*. 1–6. https://doi.org/10.1109/BESC51023.2020.9348311

[11] Benjamin Krumnow, Hugo Jonker, and Stefan Karsch. 2022. How gullible are web measurement tools? A case study analysing and strengthening OpenWPM's reliability. In *Proc. 18th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '22)*. ACM, New York, NY, USA, 16. https://doi.org/10.1145/3555050.3569131

[12] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. 2019. Tunability: importance of hyperparameters of machine learning algorithms. *J. Mach. Learn. Res.* 20, 1 (Jan. 2019), 1934–1965.

[13] Shaina Raza, Ashmal Vayani, Aditya Jain, Aravind Narayanan, Vahid Reza Khazaie, Syed Raza Bashir, Elham Dolatabadi, Gias Uddin, Christos Emmanouilidis, Rizwan Qureshi, and Mubarak Shah. 2025. VLDBench: Vision Language Models Disinformation Detection Benchmark. arXiv:2502.11361 [cs.CL] https://arxiv.org/abs/2502.11361

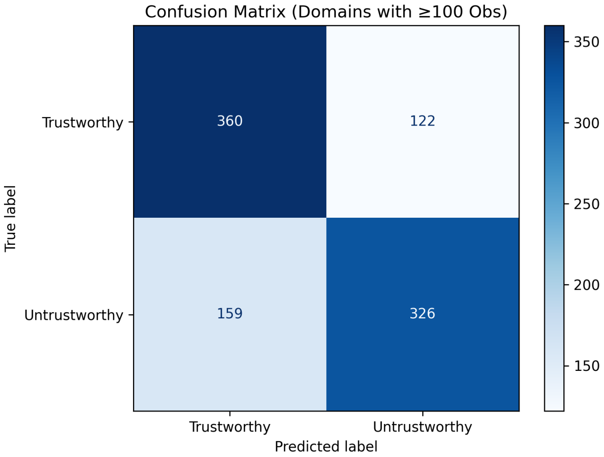

Figure 8: Confusion Matrix for the Model trained on domains with at least 100 days of observation

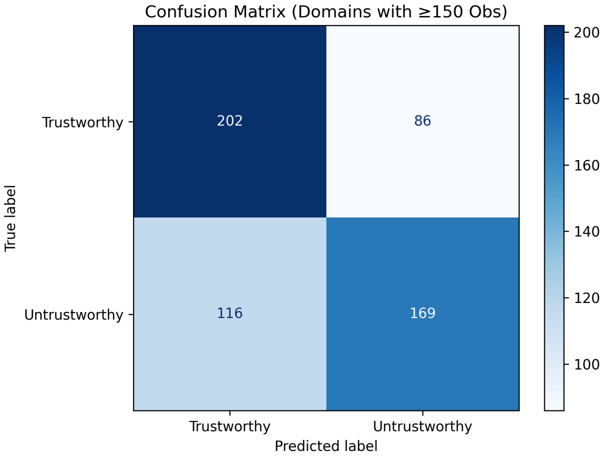

Figure 9: Confusion Matrix for the Model trained on domains with at least 150 days of observation