Toward Automated DNS Tampering Detection Using Machine Learning

Paola Calle*† Larissa Savitsky*† Arjun Nitin Bhagoji¶ Nguyen Phong Hoang§ Shinyoung Cho†
†Smith College ¶University of Chicago §University of British Columbia

ABSTRACT

DNS manipulation is one of the most prevalent and effective techniques for censoring Internet access and interfering with users' online activities worldwide. Reliable detection of DNS tampering is crucial, but challenging due to evolving censorship tactics and the lack of complete ground truth data. In this paper, we demonstrate the power of machine learning (ML) in addressing these challenges by applying supervised and unsupervised models to recent global DNS measurement data collected by the Open Observatory of Network Interference (OONI). Our models achieve high accuracy in learning expert-defined heuristics for DNS tampering and uncovering new manipulation instances missed by rule-based approaches.

Through an extensive analysis evaluating different training data volumes and time windows from one to 24 months, we provide key insights into how the quantity and diversity of data, as well as evolving censorship behaviors, impact model performance over time. Remarkably, our ML detector can enhance traditional heuristics by accurately identifying DNS fingerprints with high confidence. These findings underscore the effectiveness of ML techniques in detecting global DNS manipulation at scale while adapting to emerging censorship tactics.

To foster future research, we will release our regularly updated models, enabling the development of robust, sustainable censorship detection systems capable of withstanding the dynamic landscape of Internet censorship worldwide. Our work paves the way for more proactive interventions that safeguard Internet freedom globally.

KEYWORDS

DNS Tampering, Anomaly Detection, Machine Learning

1 INTRODUCTION

The Domain Name System (DNS) is vital for Web connectivity. DNS maps human-readable domain names to numerical IP addresses, which are necessary for routing traffic on the Internet. Due to the insecure and unauthenticated design of the traditional DNS resolution process, DNS is often targeted and abused by malicious actors [19, 22, 27]. DNS tampering occurs when DNS answers are manipulated for illicit purposes, interfering with the normal resolution process.

The DNS resolution process can be tampered with by a variety of entities such as rogue DNS resolvers [36], DNS injectors [1, 14, 23], or Internet Service Providers (ISPs) [17] or subtly disrupting Web access [22]—-to redirecting users to harmful sites for distributing malware or monetizing through ads [19, 37]. The manipulated answers often take the form of no responses [22], DNS error messages (e.g., NXDOMAIN) [4, 21, 22], private IP addresses (e.g., 127.0.0.1 [3, 6, 7, 13, 23]), or routable but wrong IP addresses [1, 13, 14, 22]. To add to the complexity, discrepancies in DNS answers can also happen for legitimate reasons such as DNS-based load balancing [31, 32]. As a result, detecting DNS manipulation is challenging since tampering signatures can vary across resolution vantage points, countries, and over time.

Prior studies have attempted to detect DNS manipulation, usually by leveraging heuristic methods [22, 27]. The basic approach involves evaluating the consistency between DNS responses observed at the testing and control sites. The latter of which is assumed to be uncensored. More specifically, IP addresses resolved for the same domain name are compared between the two sites to see if they match or belong to the same Autonomous System (AS). If they do not match, then DNS manipulation is suspected and vice versa.

Such approaches, however, could result in inaccurate inference due to the aforementioned legitimate reasons. For instance, if the control site is located in a distant location than the testing site the IP addresses resolved for the same domain name may not match due to DNS-based load balancing or sites hosted on different Content Delivery Networks (CDNs) [27, 31]. Taking advantage of such dynamic behaviors as part of normal Internet operations, sophisticated censors like China's Great Firewall (GFW) have been observed to evade detection by injecting different routable *but wrong* IP addresses to hinder straightforward detection [1, 13, 14].

The advent of machine learning (ML) has recently enabled researchers to automatically learn tampering heuristics and uncover new manipulation instances missed by hardcoded rules, effectively detecting DNS manipulation by the GFW, one of the most sophisticated censors in the world [5]. However, the applicability of these models at global scale and their effectiveness in detecting evolving tampering over time remain unknown. In this paper, we advance these efforts by applying the approach proposed previously on recent measurement data at global scale. Given the increasing advancement of machine learning and the plethora of new global measurement data, we believe that this research direction will be of interest to both networking and Internet freedom communities.

To that end, the contributions of this paper lie in our investigation of the extent to which the ML approaches are valid and applicable to global scale. We train both supervised and unsupervised models on the network measurement data collected from around the world by the Open Observatory of Network Interference (OONI) [26]. Our models achieves comparable performance in

*Co-first authors with equal contribution to this work.

Free and Open Communications on the Internet 2024 (2), 13–21

learning tampering heuristics. We also train models with incrementally expanding data windows–from one to 24 months—-to explore how increased coverage of diverse volunteer data and evolving censorship affect model accuracy and validity. Furthermore, we demonstrate how an automatic ML detector can augment heuristic approaches by identifying DNS fingerprints with high confidence, thereby improving blocking signature detection.

We will release our models and regularly update them to cope with potential changes in DNS tampering around the world and encourage future research in this domain [1].

2 BACKGROUND AND MOTIVATION

This section provides an overview of DNS manipulation and the associated detection challenges. We also explore how advancements in machine learning contribute to addressing these issues and outline the motivation for our research in demonstrating the effectiveness of ML for automated DNS tampering detection.

2.1 DNS manipulation and challenges in detection

The Domain Name System (DNS) plays a crucial role in almost every Internet communication by mapping human-readable domain names to numerical IP addresses needed for routing traffic. However, the traditional DNS resolution process is vulnerable to manipulation due to its insecure and unauthenticated design [20]. DNS manipulation occurs when DNS responses are tampered with, interfering with the normal resolution process [1, 8, 14, 28, 31, 33].

Several entities like rogue DNS resolvers, DNS injectors, or Internet Service Providers (ISPs) can carry out DNS manipulation. Their objectives range from censorship [1, 14, 23, 28, 31, 33], where users are blocked from accessing certain websites, to redirection to malicious sites for distributing malware or phishing attacks [8]. The manipulated DNS responses often take the form of no responses, DNS errors [4, 21, 22], non-routable private IP addresses [3, 6, 13, 23], or routable but incorrect IP addresses [1, 13, 14, 22]. Due to the foundational role of DNS in Web access and the ease of implementation, DNS manipulation has become one of the most common and effective methods for interfering with users' online activities, especially in repressive regions of the world.

Detecting DNS manipulation is challenging for several reasons. Firstly, DNS tampering signatures can vary across measurement vantage points, countries, and over time as censors continuously evolve their techniques [5]. For example, China's Great Firewall (GFW) has been known to dynamically change DNS tampering behaviors over the past two decades [1, 12, 14]. Furthermore, discrepancies in DNS responses can legitimately occur due to factors like DNS-based load balancing or sites hosted on different Content Delivery Networks (CDNs) [31]. Obtaining ground truth data on DNS tampering worldwide is also difficult. Collectively, these factors pose significant challenges in accurately detecting DNS manipulation at scale.

Previous studies have attempted to detect DNS manipulation by leveraging heuristic methods that evaluate the consistency between DNS responses observed at the testing and control sites [9, 22]. However, such approaches can result in inaccurate inferences due to the aforementioned legitimate reasons for discrepancies while censors continuously evolve their techniques to hamper detection and circumvention [5]. Moreover, independent confirmation of suspected manipulation can be expensive and requires a significant amount of manual effort and expertise. As a result, reliably detecting DNS manipulation requires advanced techniques that can learn complex patterns and account for the various factors influencing DNS responses.

2.2 Machine learning as an emerging approach

Given the challenges in reliably detecting DNS manipulation using rule-based heuristic methods, machine learning (ML) has recently emerged as a promising approach to tackle many security and networking problems [2, 11, 15, 29]. ML techniques have also been shown to be able to automatically learn the complex patterns and signatures of DNS tampering from data, without requiring manually defined rules [5]. This ability to discover intricate relationships makes ML a plausible solution for detecting DNS manipulation at scale, especially in the face of evolving censorship tactics.

Several recent works have applied ML models to identify DNS censorship instances missed by rule-based approaches. By training on large network measurement datasets, these ML models can learn expert-defined heuristics as well as uncover new blocking signatures. Their performance has demonstrated the effectiveness of ML in automatically detecting sophisticated DNS manipulation tactics employed by nation-state censors.

However, prior work applied ML in this context has been limited to specific countries or regions, such as China [5]. The generalizability of these ML models to new countries and evolving censorship behaviors over time remains an open question. As censors constantly adapt their techniques to avoid detection, the robustness and longevity of trained ML models needs to be evaluated.

Moreover, the lack of complete ground truth data on global DNS manipulation poses challenges in training and evaluating ML models accurately. Supervised models require high-quality labeled data, while unsupervised anomaly detection methods need careful delineation of "normal" vs "anomalous" patterns.

This work aims to advance the application of ML for detecting DNS manipulation at global scale over time. By replicating previous approaches on recent worldwide network measurement data, we investigate the portability and sustainability of ML models for this problem. Our techniques for curating training data and addressing sampling biases can inform future efforts. Ultimately, our goal is to develop robust, regularly-updated ML models that can reliably identify evolving DNS tampering tactics used by censors around the world.

3 DATA COLLECTION AND MACHINE LEARNING PIPELINE

This section describes the dataset used in our study, the rationale behind our choice of this dataset, and the machine learning pipeline we developed to detect global DNS tampering. Figure 1 illustrates our machine learning pipeline from data collection to detection of DNS tampering using the OONI dataset.

[1] https://github.com/InternetCensorship/dns_ml_scripts

3.1 Dataset

Our study utilizes data from OONI [9], which is one of the earliest efforts to monitor global Internet censorship, offering both raw data and labels essential for our analysis. OONI stands out as the sole platform providing comprehensive and up-to-date dataset with labels on global censorship at the time of our study. Initially, we also intended to use Satellite dataset [31] from the Censored Planet project [34], but recent changes in the format of their dataset and a lack of labeled data limited our capacity to use this resource. As detailed in §4, the availability of labeled data is essential for the evaluation of our ML models.

OONI's efforts in monitoring Internet censorship contribute significantly to our understanding of network interference, offering insights into network performance and censorship incidents. With an extensive collection of over 1.7 billion measurements from more than 200 countries as of the time of our study, OONI's dataset is invaluable for training and testing our models [26]. However, leveraging this vast dataset also presents several challenges, particularly in data engineering, to make it amenable to machine learning (ML) applications. We discuss these challenges and our approaches to address them in the following subsection.

Initial dataset. Our initial dataset comprises OONI probes collected over a two-year period, from January 1, 2022, to December 31, 2023, leveraging OONI's "blocking" label to train supervised models and evaluate the accuracy of unsupervised models. It is important to note that obtaining ground truth for DNS tampering is challenging, and OONI's "blocking" label may contain false positives flagged as anomalies [25]. We get multiple models, both supervised and unsupervised, trained on the same dataset to explore the differences between OONI's labels and the models' output. This approach allows us to comprehensively understand the complex nature of DNS tampering and identify critical factors that influence anomaly detection.

3.2 Data cleaning and preprocessing

Ensuring the high quality of training datasets is crucial for the effectiveness of machine learning models, as it directly influences their ability to identify patterns accurately. This is particularly true in DNS tampering detection, where a DNS failure could indicate tampering, requiring distinguishing from measurement failures caused by generic network errors or issues with the client device. This is even more critical for unsupervised models that determine underlying structures without labeled data. To improve the quality of the OONI dataset, we sanitize the data by removing records with missing values and failed measurements, normalize the data formats, and convert categorical variables into numerical representations through one-hot encoding.

Table 1 provides a summary of the initial dataset and the preprocessed dataset, including the number of records labeled as DNS censorship or not.

Dropping invalid records. To ensure high quality of training datasets, we set up criteria for the inclusion of records from OONI's dataset. Recognizing that normal operational failures–not necessarily related to censorship–can lead to missing or invalid data, we filter out such records. Specifically, we exclude records with incomplete or incorrect probe ASN and resolver ASN, which represent the ASN information of the measurement location and the DNS resolver used, respectively. These features are crucial for the effective training of our model. In addition, records with an undetermined body proportion are removed to maintain the dataset's accuracy, as this value is also used for training. Finally, we exclude the country code from our dataset as this categorical variable is not used in training the model.

Duration	Jan. 1, 2022 - Dec. 31, 2023
Initial dataset	888M
Pre-processed dataset	800M
Clean	766M
Anomalous DNS	5.20M
Pre-encoding Features	14
Encode Features	73

Table 1: Number of records and features in our curated datasets.

To tailor our datasets specifically for DNS anomaly detection, we retain only records marked as "accessible" or "DNS tampered." This process ensures the dataset's high quality. Table 1 illustrates the number of data retained after applying these filters, confirming the adequacy of our datasets for training our ML models.

Filling missing fields in the control datasets. In our ML models, one of the important features is the matching between the ASNs to which the IP addresses of the DNS response belong and the ASNs of the IP addresses received by the control node. However, not all OONI's control records contain IP-to-ASN data, as this was an enhancement integrated into the OONI platform starting in 2023. To adapt our models for analysis of data points collected prior to this enhancement, we fill the missing fields in the control datasets, which are crucial for our model training. To accomplish this, we utilize services from Team Cymru [35] and IPinfo.io [16].

Team Cymru offers historical IP to ASN lookup services and supports bulk queries, enabling us to retrieve precise ASN information corresponding to the time when the data was collected. However, occasionally, it returns multiple ASNs for a single IP address, which could be attributed to various factors such as "multihoming" or "sub-allocating". In the case where multiple ASNs are returned, we utilize IPinfo.io to determine the most appropriate ASN from the available options. IPinfo.io does not provide historical data, which is the primary reason for not using it exclusively across all experiments. Instead, it serves as a complement to Team Cymru, contributing to the enhancement of our control dataset's accuracy.

Classifying "clean" and "anomalous" data. In the OONI dataset, a record is classified as "clean" if it is not marked as "DNS tampered" by OONI and if the IP addresses it returns are consistent with those observed by OONI's control node. Records that do not meet these criteria are labeled as "anomalous." In our unsupervised learning models, we train the algorithms using data labeled as "normal" (i.e., clean), while acknowledging the potential for rare and minimal false positives [25], assuming that the data is unlikely to have DNS tampering instances. In our supervised learning models, the training datasets include a mix of both "clean" and "anomalous" records.

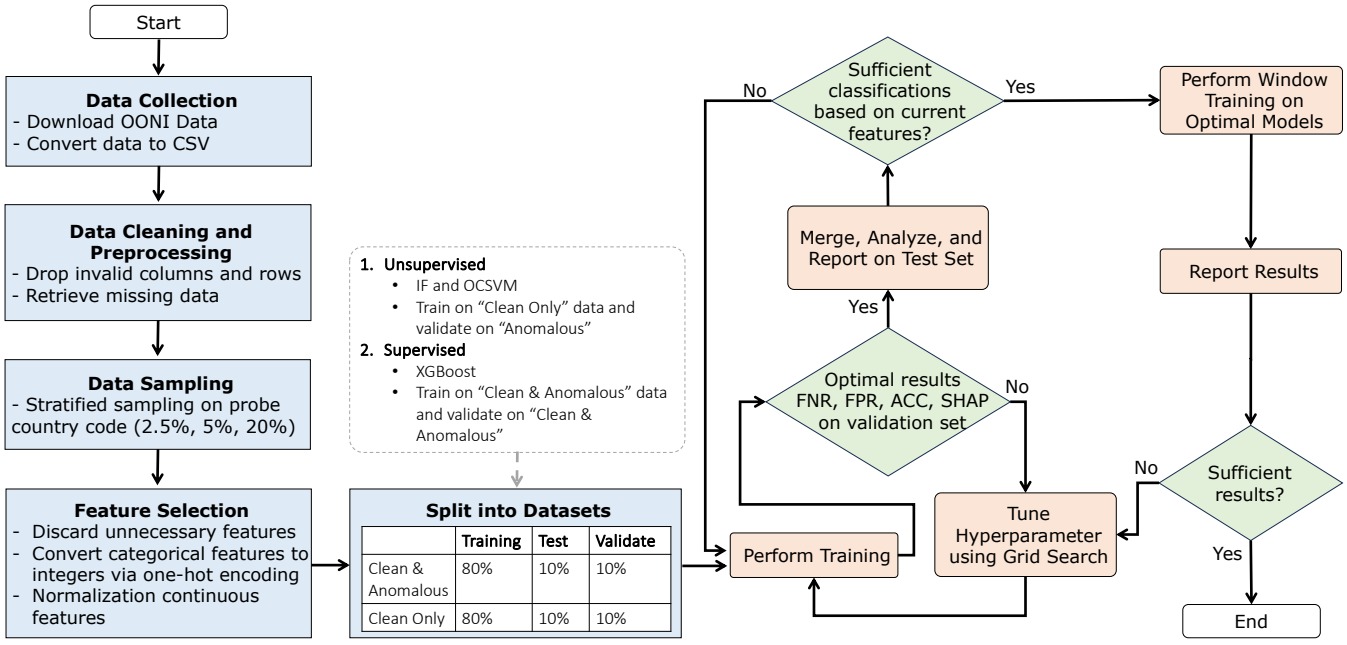

Figure 1: Our ML pipeline from data collection to DNS anomaly detection using OONI dataset.

3.3 Addressing biases with data sampling and balancing

Despite OONI's extensive global coverage, significant disparities exist in measurement volumes across countries due to reliance on volunteer contributions. For instance, by the end of 2023, the US had a measurement count that was 28 times higher than that of Turkey. This difference can be attributed to OONI's reliance on volunteer contributions for data collection, with certain regions facing challenges in volunteer recruitment.

To address these disparities and ensure a balanced analytical perspective, we implement a stratified random sampling approach to avoid any potential biases resulting from uneven sample distribution and provide a representative global dataset. This approach divides the data into subgroups (*i.e.* strata) based on the country code of each probe. We then draw a proportional, random sample from each stratum, thereby ensuring that our sample accurately mirrors the characteristics of each subgroup and the collective dataset.

To mitigate sampling biases, we train and evaluate our models using different sample sizes of 2.5%, 5%, and 20% to better identify potential issues arising from varying sample sizes

3.4 Feature Selection

Initial selection. The OONI dataset comprises various features. To mitigate the risk of over-fitting, where the model could learn irrelevant details rather than underlying patterns, we exclude IP addresses and domains from our training set. This decision is based on the limited shared IP address records, which could lead the model to prioritize noise over meaningful patterns. Our model includes both resolver ASN and probe ASN to account for topological factors of the client and resolver. Table 2 presents the complete set of features retained for our training.

Feature scaling. For categorical variables, we employ one-hot encoding to convert them into numerical representations. This involves assigning a unique numeric identifier to each distinct level within a variable, facilitating one-hot encoding. For variables like http experiment failure and dns experiment failure, which are in string format, we group the strings into predefined error categories before applying one-hot encoding.

Unlike other variables, for the ASNs related to both probes and resolvers, we apply the encoding without assigning unique labels to each ASN. Instead, we directly encode them to retain structural or hierarchical information inherent in the ASNs.

ASN control match. In addition, we also introduce a derived feature, "ASN control match", to offer a looser interpretation of whether ASNs differ between the control and testing sites. This feature is set to 1 if there is any overlap between the resolved ASN and control ASN, indicating at least one common ASN.

3.5 Training

Dataset curation. We partition the dataset into three separate sets: training (80%), validation (10%), and testing (10%). When training supervised models, we combine "clean" and "anomalous" data in each set. However, for unsupervised models, we use only "clean" data for training, while both "clean" and "anomalous" data are included in the validation and test sets. The rationale behind this approach is to ensure that the model is trained on normal data, which is more abundant, while also being exposed to anomalies during validation and testing to evaluate its performance in detecting DNS tampering. We use the validation set to choose hyper-parameters, settings that

Table 2: Description of the features used to train our machine learning models

Feature Name	Data Type	One-Hot Encoded	Derived	Description
probe_network_name	Discrete	Y	N	Network name where the testing client is located
resolver_network_name	Discrete	Y	N	Network name of DNS resolver used by testing client
domain_name	Discrete	Y	N	Domain being tested during experiment
dns_experiment_failure	Discrete	Y	N	Indicates failure in DNS experiment
http_experiment_failure	Discrete	Y	N	Indicates HTTP connection failure despite DNS consistency
status_code_match	Discrete	Y	Y	Checks HTTP status code match at client and control server
headers_match	Discrete	Y	Y	Compares HTTP headers at client with those at control server
title_match	Discrete	Y	Y	Verifies webpage title match at client and control server
probe_asn	Discrete	Y	N	Testing client ASN
resolver_asn	Discrete	Y	N	ASN of DNS resolver used by testing client
test_runtime	Continuous	N	N	Total runtime of test, in seconds
measurement_start_time	Continuous	N	N	Start time of the measurement
asn_control_match	Discrete	N	Y	Checks if DNS response ASN matches control ASN
body_proportion	Continuous	N	Y	Measures proportionality between control and response body

guide the model's learning, and held-out tests, using new data to check the model's performance, for overall performance evaluation.

Iterative feature selection. For our final feature set determination, we utilize an iterative process where models were trained on various combinations of explanatory variable sets. Our best models include both resolver ASN and probe ASN, resulting in higher reliability. We analyze their FNR (False Negative Rate), FPR (False Positive Rate), ACC (Accuracy), SHAP (Shapley Additive exPlanations) values to understand each feature's impact. Based on these insights, we update the feature set in our training dataset to optimize for both interpretability and accuracy in our model's performance.

3.6 Time window

To examine how the training dataset size and time span affect model performance, we adopt an expanding window evaluation method. We begin with a training set comprising 2.5% of the data from December 2023 and incrementally expand backward in time, adding data from each month until we reach January 2022. This approach allows us to determine whether larger window sizes lead to better anomaly detection accuracy and evaluate the coverage needed for our models to maintain their usefulness and accuracy.

3.7 ML algorithms

We adopt two unsupervised models–Isolated Random Forest (IF) [18] and the One-Class Support Vector Machine (OCSVM) [30]– and one supervised model, XGBoost (XGB) [10]. Each model undergoes a detailed tuning process where we adjust and test various parameters to optimize performance.

Isolation Random Forest (IF) [18]. This unsupervised model, constructed with decision trees similar to Random Forests, does not need predefined labels, which is advantageous for anomaly detection, particularly in situations like DNS manipulation where ground truth is not trivial to obtain. It effectively isolates outliers by dividing the data using chosen features, making it highly effective for pinpointing unusual data points.

One-Class Support Vector Machine (OCSVM) [30]. OCSVM is adept at detecting unusual patterns or new occurrences, which can enhance its effectiveness in identifying DNS manipulation. This is because OCSVM employs a different strategy from tree-based models like Isolation Forest (IF) by trying to delineate a clear boundary between normal data and outliers in a high-dimensional space. While both OCSVM and IF analyze all available features, the manner in which they do so differs, with OCSVM using a hyperplane for separation and IF using decision trees to isolate points, potentially making OCSVM more suited for certain types of anomaly detection.

XGBoost (XGB) [10]. This is a supervised learning method that combines multiple decision trees to make more accurate predictions, a strategy known as ensemble learning. It's part of the gradient boosting family, where models are built sequentially to correct the errors of previous ones. XGBoost also uses regularization, a technique to prevent the model from fitting too closely to the training data, thereby enhancing its ability to perform well on unseen data.

4 ANALYSIS AND FINDINGS

In this section, we present a comprehensive analysis of our machine learning models' performance in detecting global DNS manipulation. We evaluate various aspects, including overall accuracy, false positive and negative rates, and feature importance. Additionally, we investigate the impact of training data size and time window on model effectiveness. Finally, we highlight the capability of our models to discover blocking signatures and support existing ones, ultimately enhancing our understanding of evolving DNS tampering tactics worldwide.

Table 3: Comparative performance metrics across all our machine learning models (sample size=2.5%)

Model	FNR	FPR	TPR	TNR	ACC
IF	0.0718	0.1321	0.9282	0.8679	0.8699
OCSVM	0.0244	0.9711	0.9756	0.0289	0.0598
XGB	0.0597	0.0005	0.9403	0.9995	0.9991

4.1 Performance Evaluation

Data sampling. To mitigate potential biases arising from uneven sample distribution across countries, we employ a stratified random sampling approach. This technique ensures that our training and evaluation datasets accurately represent the characteristics of each subgroup (country) and the collective global dataset. We train and evaluate our models using stratified random samples of 2.5%, 5%, and 20% of the full dataset. Experimental results reveal no significant performance differences across these sample ratios, with the XGBoost (XGB) model exhibiting minimal variance in performance metrics. Consequently, we select a 2.5% sample size for computational efficiency.

Model performance comparison. Table 3 compares the performance metrics for our three machine learning models (IF, One-Class SVM, and XGB) using a 2.5% sample size. As expected for a supervised approach, XGB outperforms the other models, achieving the lowest False Positive Rate (FPR) of 0.0005, indicating strong agreement with OONI's blocking labels with high confidence. The IF model demonstrates commendable overall performance, effectively balancing True Positives and True Negatives. In contrast, the One-Class SVM exhibits notably lower accuracy coupled with a low True Negative Rate (TNR) but higher True Positive Rate (TPR) and FPR, indicating a bias towards predicting instances as censored. An in-depth analysis of One-Class SVM's performance characteristics is scope for future investigation.

Feature importance analysis. Understanding the significance of features in a model's decision-making process is crucial for interpreting its behavior and the influence of each factor. Table 4 highlights the top 10 most important features for the IF and XGB models, as measured by SHAP (SHapley Additive exPlanations) values. The IF model assigns the highest importance to the resolver ASN and probe ASN features, underscoring the relevance of ASN information. Conversely, for XGB, 'headers match' and 'asn control match' prove to be more critical indicators, highlighting a divergence in feature importance between the models.

4.2 Performance Window

We investigate how the size of the training dataset affects the performance of our models over time. Specifically, we examine how much monthly data is required over the two-year period to create an accurate predictive model that can generalize to multiple months beyond the training period. Figure 2 displays the distribution of records in the OONI dataset, labeled as either "accessible" or experiencing "DNS tampering."

Our analysis reveals that the Isolation Forest (IF) model performs best when trained on one to six months of data. Remarkably, for the XGBoost (XGB) model, we find that using only a single month's worth of data is sufficient to achieve optimal predictive performance.

Table 4: Top 10 features importance for IF and XGB

Feature	IF	XGB
ASN Control Match	0.025 [6]	0.177 [3]
Body Proportion	0.017 [7]	0.087 [5]
Domain Name	0.123 [4]	0.051 [6]
Headers Match	0.009 [9]	0.317 [1]
HTTP Experiment Failure	0.000 [10]	0.118 [4]
Probe ASN	0.220 [2]	0.032 [7]
Probe Network Name	0.148 [3]	0.023 [9]
Resolver ASN	0.237 [1]	0.021 [10]
Resolver Network Name	0.169 [5]	0.023 [8]
Status Code Match	0.012 [8]	0.097 [2]

This finding highlights the potential of machine learning models to extend their predictions well beyond the initial training period while maintaining high accuracy levels and low false negative and false positive rates. The graph in Figure 3 (a) exhibits the performance of the IF model trained on different amounts of data. Notably, the model with the lowest false negative rate (FNR) is the one trained on two months of data. Furthermore, the figure demonstrates a general decrease in the false positive rate (FPR) as we increase the number of months in the dataset, with the FPR reaching its minimum point at 24 months. However, this trend comes at the expense of an increasing FNR, which rises at a greater rate as the number of months grows. Lastly, the figure illustrates that the model trained on a smaller dataset, one month in duration, achieved the highest accuracy, and accuracy tends to drop 5 percentage points (PP) as more months are added. One possible explanation for these results may be over-fitting.

In contrast, Figure 3 (b) illustrates that the performance fluctuations of the XGB model are comparatively subtle. The deviations from the lowest score range from 0 to 5 PP for the FNR, 0.07 PP for the FPR, and 0.10 PP for Accuracy (ACC). Additionally, models trained with shorter duration of data typically exhibit higher accuracy, with peaks observed at one month and four months. These observations imply that although larger training datasets may capture a broader spectrum of patterns, they also have the potential to introduce noise and contribute to overfitting, which could reduce the model's overall effectiveness.

Our analysis demonstrates the importance of carefully selecting the appropriate training dataset size and time window to achieve optimal performance in detecting DNS tampering using machine learning models. It also highlights the potential of these models to generalize beyond their initial training period, providing a promising direction for developing robust and sustainable censorship detection systems.

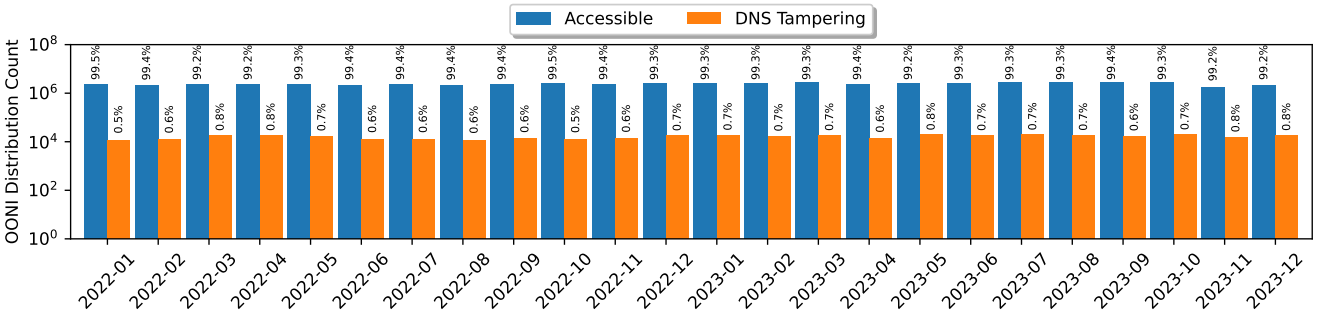

Figure 2: Distribution count of records in the OONI dataset, labeled as either accessible or indicative of DNS tampering.

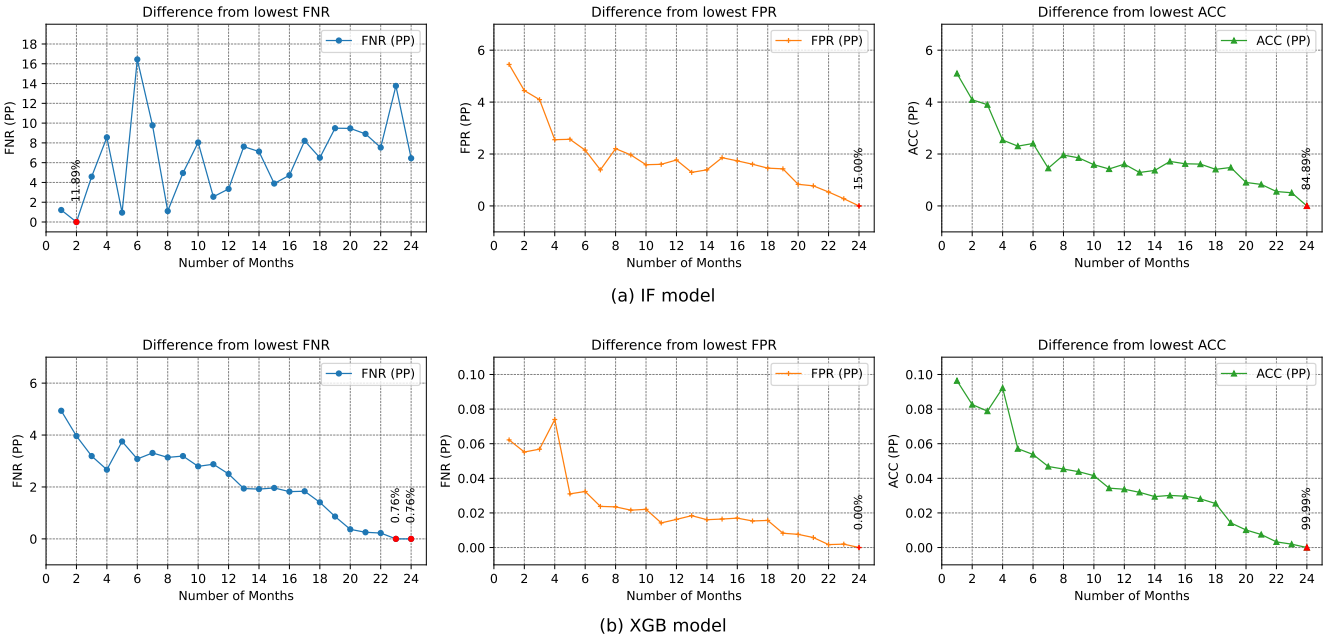

Figure 3: FNR, FPR, and ACC distributions as the training set cumulatively increase by one month in each model. (The red-colored markers indicates the lowest values.)

4.3 Signature Discovery

A key objective of our study is to explore whether machine learning (ML) models can uncover new blocking signatures beyond those already identified by OONI's existing rules [24]. To achieve this, we develop a robust methodology that leverages multiple models within each ML approach to generate reliable predictions through a majority voting scheme.

Specifically, we allow the individual models within each ML setup (e.g., three models for OCSVM) to "vote" on their predictions, and we accept the decision made by the majority. This majority rule approach is complemented by considering confidence scores to ensure the reliability of our predictions. We then evaluate all records that both OONI's labels and our majority prediction (supported by high confidence) classify as censored cases—these represent true positive instances.

To mitigate potential false positives, we exclude records with fewer than 50 similar instances, applying a conservative threshold for further analysis. Subsequently, we investigate records not included in OONI's predefined signatures, focusing on those that meet our criteria. Our goal is to confidently identify new DNS censorship fingerprints that can be added to the existing rule set.

The flexibility of unsupervised models like OCSVM and IF is vital for pattern recognition, as these models can detect a wide range of DNS censorship patterns without relying on pre-labeled data, unlike the supervised XGBoost model, which depends on known labels. Table 5 presents our findings, showcasing the potential of ML models in identifying DNS censorship instances while highlighting the necessity of manual verification to address potential misclassifications, such as confusing load balancing for blocking.

Our methodology not only highlights the potential of ML models in identifying DNS censorship but also underscores the importance of combining ML analysis with manual review. This combined approach effectively enhances our ability to validate and discover new censorship patterns, reinforcing the robustness of our approach.

By leveraging the strengths of both unsupervised and supervised ML models, along with manual verification, we demonstrate a comprehensive strategy for uncovering evolving DNS tampering tactics employed by censors worldwide. Our findings pave the way for developing more dynamic and precise interventions in the landscape of global Internet censorship. We are planning to share these fingerprints that our models have identified with high confidence with the OONI community to enhance their existing rule set and improve the usability of the OONI Explorer tool by increasing the number of anomalous probes that can be marked as "confirmed" censorship events.

5 RELATED WORK

DNS censorship is a prevalent form of network interference that restricts access to specific websites by manipulating DNS responses. Detecting DNS manipulation is crucial for understanding the extent of censorship and its impact on Internet freedom. Numerous efforts have been made to develop techniques for detecting DNS censorship [22, 28, 31], mostly focusing on rule-based methods that rely on expert-defined heuristics to identify anomalies in DNS responses. However, as discussed earlier in the paper, these methods are limited in their ability to adapt to new censorship techniques and evolving tampering strategies.

Recently, with the increasing availability of large network measurement datasets and advancements in machine learning techniques, researchers have explored the application of ML models to detect DNS censorship. Brown et al. [5] propose a machine learning approach to automate the detection of DNS manipulation. The authors train supervised and unsupervised models on network measurement data collected by OONI [9] and Censored Planet [34] in China and the United States. Evaluating the models using the state-of-the-art labeled censored domain names detected by GFWatch [14]. This seminal work demonstrates the potential of ML models in detecting DNS censorship and highlights the importance of leveraging large-scale network measurement datasets for training and evaluating these models. Nevertheless, the generalization of the proposed approach to other countries and the effectiveness of the models in detecting evolving tampering over time remain unclear, motivating our work to tackle broader research questions of whether these proposed ML methods can be applied at a global scale and how they perform in detecting DNS manipulation in different countries and over time.

6 CONCLUSION

In this study, we demonstrated the powerful capability of machine learning models to detect global DNS tampering at scale. Our supervised and unsupervised models achieved high accuracy by learning expert-defined heuristics and uncovering new censorship instances missed by rule-based approaches, showcasing their effectiveness against evolving manipulation tactics.

Table 5: Signature Discovery from OCSVM and IF Models

Counts	Injected IP	Known Fingerprint	Country
52	195.19.90.226	False	RU
53	202.169.44.80	True	ID
56	49.205.171.201	True	IN
56	173.209.39.114	True	CA
57	208.91.112.55	True	Multiple
58	171.25.175.70	True	RU
61	95.167.13.51	False	RU
61	95.167.13.50	False	RU
61	188.19.132.154	False	RU
61	188.19.132.155	False	RU
63	78.128.216.33	False	CZ
64	90.207.238.183	False	GB
66	90.255.255.14	False	GB
69	80.250.8.1	False	CZ
72	217.175.53.72	True	IT
77	218.248.112.60	True	IN
79	81.200.2.238	True	RU
104	83.224.65.79	False	IT
126	94.140.14.35	False	US and CA
144	85.142.29.248	False	RU
167	36.86.63.185	True	ID
169	83.224.65.74	True	IT
231	49.44.79.236	True	IN
242	188.186.154.88	True	RU
261	202.83.21.14	True	IN
281	*13.127.247.216	False	IN
288	167.233.14.14	False	DE
305	188.186.146.208	True	RU
311	188.186.146.207	True	RU
338	188.186.154.79	True	RU
383	175.139.142.25	True	MY
468	195.175.254.2	True	TR
662	146.112.61.106	True	Multiple

Note: An asterisk (*) before an IP address indicates that the IF model also identified this record.

Through evaluations spanning 1 to 24 months of training data, we gained insights into how data quantity, diversity, and the dynamics of censorship impact model performance over time. Notably, our models generalized well beyond the initial training period while maintaining high accuracy.

A key highlight was our automatic ML detector that accurately identified DNS fingerprints, including previously undocumented censorship patterns. This underscores machine learning's potential to drive proactive interventions safeguarding Internet freedom globally.

While promising, continuous monitoring and model updates are crucial due to evolving censorship landscapes. We will release our regularly updated models to foster further research in developing robust, sustainable censorship detection systems worldwide.

REFERENCES

[1] Anonymous, Arian Akhavan Niaki, Nguyen Phong Hoang, Phillipa Gill, and Amir Houmansadr. 2020. Triplet Censors: Demystifying Great Firewall's DNS Censorship Behavior. In *Proceedings of the 10th USENIX Workshop on Free and Open Communications on the Internet*.

[2] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, and K. Rieck. 2014. Drebin: Effective and explainable detection of android malware in your pocket.. In *NDSS*.

[3] Simurgh Aryan, Homa Aryan, and J. Alex Halderman. 2013. Internet Censorship in Iran: A First Look. In *Free and Open Communications on the Internet*. USENIX.

[4] S. Bortzmeyer and S. Huque. 2016. *NXDOMAIN: There Really Is Nothing Underneath*. RFC 8020. IETF. https://tools.ietf.org/html/rfc8020

[5] Jacob Brown, Xi Jiang, Van Tran, Arjun Nitin Bhagoji, Nguyen Phong Hoang, Nick Feamster, Prateek Mittal, and Vinod Yegneswaran. 2023. Augmenting Rule-based DNS Censorship Detection at Scale with Machine Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3750–3761.

[6] Abdelberi Chaabane, Terence Chen, Mathieu Cunche, Emiliano De Cristofaro, Arik Friedman, and Mohamed Ali Kaafar. 2014. Censorship in the Wild: Analyzing Internet Filtering in Syria. In *Internet Measurement Conference*. ACM.

[7] Michelle Cotton, Leo Vegoda, and Ron Bonica. 2010. *Special Use IPv4 Addresses*. RFC 5735. Internet Engineering Task Force. Accessed: 2024.

[8] Haixin Duan, Nicholas Weaver, Zongxu Zhao, Meng Hu, Jinjin Liang, Jian Jiang, Kang Li, and Vern Paxson. 2012. Hold-on: Protecting against on-path DNS poisoning. In *Proc. Workshop on Securing and Trusting Internet Names, SATIN*. Citeseer.

[9] Arturo Filastò and Jacob Appelbaum. 2012. OONI: Open Observatory of Network Interference. In *Free and Open Communications on the Internet*. USENIX.

[10] JH. Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001).

[11] G. Ho, A. Cidon, L. Gavish, M. Schweighauser, V. Paxson, S. Savage, GM. Voelker, and G. Wagner. 2019. Detecting and characterizing lateral phishing at scale. In *USENIX Security*.

[12] Nguyen Phong Hoang, Jakub Dalek, Masashi Crete-Nishihata, Nicolas Christin, Vinod Yegneswaran, Michalis Polychronakis, and Nick Feamster. 2024. GFWeb: Measuring the Great Firewall's Web Censorship at Scale. In *USENIX Security Symposium*. USENIX.

[13] Nguyen Phong Hoang, Sadie Doreen, and Michalis Polychronakis. 2019. Measuring I2P Censorship at a Global Scale. In *Proceedings of the 9th USENIX Workshop on Free and Open Communications on the Internet*.

[14] Nguyen Phong Hoang, Arian Akhavan Niaki, Jakub Dalek, Jeffrey Knockel, Pellaeon Lin, Bill Marczak, Masashi Crete-Nishihata, Phillipa Gill, and Michalis Polychronakis. 2021. How Great is the Great Firewall? Measuring China's DNS Censorship. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security)*. 3381–3398.

[15] A. Hounsel, J. Holland, B. Kaiser, K. Borgolte, N. Feamster, and J. Mayer. 2020. Identifying disinformation websites using infrastructure features. In *FOCI*.

[16] IPInfo.io. 2024. IPinfo. https://ipinfo.io/

[17] Joongyum Kim, Junghwan Park, and Sooel Son. 2021. The Abuser Inside Apps: Finding the Culprit Committing Mobile Ad Fraud. In *Proceedings of the Network and Distributed System Security (NDSS) Symposium*.

[18] FT. Liu, KM. Ting, and Z. Zhou. 2008. Isolation forest. In *ICDM*.

[19] Minzhao Lyu, Hassan Habibi Gharakheili, and Vijay Sivaraman. 2022. A Survey on DNS Encryption: Current Development, Malware Misuse, and Inference Techniques. *Comput. Surveys* 55, 8 (2022), 1–28.

[20] P. Mockapetris. 1987. *Domain Names - Concepts And Facilities*. RFC 1034. IETF. https://datatracker.ietf.org/doc/html/rfc1034

[21] Z. Nabi. 2013. The Anatomy of Web Censorship in Pakistan. In *FOCI '13*.

[22] Arian Akhavan Niaki, Shinyoung Cho, Zachary Weinberg, Nguyen Phong Hoang, Abbas Razaghpanah, Nicolas Christin, and Phillipa Gill. 2020. ICLab: A global, longitudinal internet censorship measurement platform. In *Proceedings of the IEEE Symposium on Security and Privacy (SP)*. 135–151.

[23] Sadia Nourin, Van Tran, Xi Jiang, Kevin Bock, Nick Feamster, Nguyen Phong Hoang, and Dave Levin. 2023. Measuring and Evading Turkmenistan's Internet Censorship. In *The International World Wide Web Conference (WWW '23)*.

[24] Open Observatory of Network Interference (OONI). 2024. Blocking Fingerprints: DNS. GitHub repository. https://github.com/ooni/blocking-fingerprints/blob/main/fingerprints_dns.csv

[25] OONI. 2022. FQA: What do you mean by Anomalies? https://ooni.org/support/faq/#what-do-you-mean-by-anomalies

[26] OONI Explorer. 2012. Uncover Evidence of Internet Censorship Worldwide. https://explorer.ooni.org

[27] Paul Pearce, Ben Jones, Frank Li, Roya Ensafi, Nick Feamster, Nick Weaver, and Vern Paxson. 2017. Global measurement of DNS manipulation. In *Proceedings of the 26th USENIX Security Symposium (USENIX Security)*. 307–323.

[28] P. Pearce, Ben Jones, F. Li, Roya Ensafi, N. Feamster, N. Weaver, and V. Paxson. 2017. Global Measurement of DNS Manipulation. In *USENIX Security Symposium*.

[29] B. Polverini and W. Pottenger. 2011. Using clustering to detect Chinese censorware. In *CSIIRW*.

[30] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural computation* 13, 7 (2001), 1443–1471.

[31] W. Scott, T. Anderson, T. Kohno, and A. Krishnamurthy. 2016. Satellite: Joint Analysis of CDNs and Network-Level Interference. In *ATC '16*.

[32] Anees Shaikh, Renu Tewari, and Mukesh Agrawal. 2001. On the effectiveness of DNS-based server selection. In *Proceedings of the IEEE INFOCOM*, Vol. 3. 1801–1810.

[33] Sparks and Neo and Tank and Smith and Dozer. 2012. The Collateral Damage of Internet Censorship by DNS Injection. *SIGCOMM CCR '12* (2012).

[34] Ram Sundara Raman, Prerana Shenoy, Katharina Kohls, and Roya Ensafi. 2020. Censored Planet: An Internet-wide, Longitudinal Censorship Observatory. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 49–66.

[35] Team Cymru. 2024. Team Cymru. https://www.team-cymru.com/

[36] Martino Trevisan, Idilio Drago, Marco Mellia, and Maurizio M Munafo. 2017. Automatic Detection of DNS Manipulations. In *Proceedings of the 2017 IEEE International Conference on Big Data (Big Data)*. 4010–4015.

[37] Nicholas Weaver, Christian Kreibich, and Vern Paxson. 2011. Redirecting DNS for Ads and Profit. In *Workshop on Free and Open Communications on the Internet (FOCI)*.