PREMADOMA: An Operational Solution for DNS Registries to Prevent Malicious Domain Registrations

Jan Spooren
jan.spooren@cs.kuleuven.be
imec - DistriNet, KU Leuven
Belgium

Thomas Vissers
thomas.vissers@cs.kuleuven.be
imec - DistriNet, KU Leuven
Belgium

Peter Janssen
peter.janssen@eurid.eu
EURid VZW
Belgium

Wouter Joosen
wouter.joosen@cs.kuleuven.be
imec - DistriNet, KU Leuven
Belgium

Lieven Desmet
lieven.desmet@cs.kuleuven.be
imec - DistriNet, KU Leuven
Belgium

ABSTRACT

DNS is one of the most essential components of the Internet, mapping domain names to the IP addresses behind almost every online service. Domain names are therefore also a fundamental tool for attackers to quickly locate and relocate their malicious activities on the Internet. In this paper, we design and evaluate PREMADOMA, a solution for DNS registries to predict malicious intent well before a domain name becomes operational. In contrast to blacklists, which only offer protection after some harm has already been done, this system can prevent domain names from being used before they can pose any threats. We advance the state of the art by leveraging recent insights into the ecosystem of malicious domain registrations, focusing explicitly on facilitators employed for bulk registration and similarity patterns in registrant information. We thoroughly evaluate the proposed prediction model's performance and adaptability on an 11 month testing set, and address complex and domain-specific dataset challenges. Moreover, we have successfully deployed PREMADOMA in the production environment of the *.eu* ccTLD registry to detect and prevent malicious registrations, and have contributed to the take down of 58,966 registrations in 2018.

CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection and malware mitigation**; • **Networks → Naming and addressing**; • **Information systems →** *World Wide Web.*

KEYWORDS

Domain Name Registration, Early Detection, Malicious Domains

ACM Reference Format:
Jan Spooren, Thomas Vissers, Peter Janssen, Wouter Joosen, and Lieven Desmet. 2019. PREMADOMA: An Operational Solution for DNS Registries to Prevent Malicious Domain Registrations. In *2019 Annual Computer Security Applications Conference (ACSAC '19), December 9–13, 2019, San Juan, PR, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3359789.3359836

ACSAC '19, December 9–13, 2019, San Juan, PR, USA

ACM ISBN 978-1-4503-7628-0/19/12…$15.00
https://doi.org/10.1145/3359789.3359836

1 INTRODUCTION

Domain names remain a major facilitator of cyberattacks. Malicious actors continuously deploy domains in their cybercriminal operations, such as spam, phishing, malware distribution and botnet C&C. Due to this crucial role in cybercriminal operations, stopping malicious domain names has become a highly important security objective.

The most well-known countermeasure for malicious domains is a *blacklist*. So-called *reputation providers* curate lists of domain names that are associated with internet-based attacks. Various software and services consult these blacklists and block incoming or outgoing communication with listed domains accordingly. Blacklists have become more agile and nowadays domain names are blocked quickly after exhibiting attacking behavior.

In response, miscreants have adopted hit-and-run strategies. They counter the short usable lifespan of their domain names by registering batches of disposable "*burner domains*" to sustain their malicious operations, resulting in large-scale registration *campaigns*. [10, 22]. Therefore, post-factum detection, such as by blacklists, is becoming limited in its effects [13].

This situation expresses the need to block malicious domain registrations before they are able to execute any attacking behavior. Hence, more recent security research aims to shift to earlier detection of malicious domain names. In a ground-breaking paper, Hao et al. [9] proposed to predict the maliciousness of domain names *at the time of registration*, using a set of 22 manually crafted features derived from data available at registration time and a *Convex Polytope Machine* (CPM) classifier.

Subsequent work by Vissers et al. [22], showed that in the .eu *top level domain* (TLD), approximately 80% of malicious domain registration campaigns are registered by maximum 20 actors, individually using very different modi operandi. This could also explain why the detection accuracy reported by Hao et al. [9] for the .net TLD (61%) differs significantly from the .com TLD (70%) at the same FPR: Different TLDs will likely have different sets of malicious actors, with different operational characteristics, yielding different detection results. Moreover, the actual operational deployment of such a detection system in a real and live environment drastically

changes this ecosystem, as malicious actors actively adapt their strategies.

In this paper, we propose, implement and thoroughly evaluate Premadoma, a security system for DNS registries that is able to detect malicious domains at registration time. We advance the state-of-the-art as set by Predator [9] by incorporating the *registrant data*, which has not been done before and by using a combination of *clustering* for making similarity-based predictions, as well as traditional machine learning *classification* for performing reputation-based classification. Finally, we strongly focus on *real-world operational aspects*, since Premadoma has effectively been deployed at the *.eu* ccTLD.

1.1 Strategy

The general goal of the Premadoma system is to reduce the amount of cybercriminal operations by detecting and preventing malicious domains at registration time. By applying an automated and adaptive mitigation strategy based on insights of the malicious domain ecosystem, Premadoma aims to substantially increase the cost for attackers and disincentivize malicious actors from launching campaigns.

1.1.1 Economic disincentivization. The registration of domain names for malicious purposes has a cost beyond the pure cost of the domain name registration: Malicious actors must provide valid-looking, in some cases functional (or at least consistent) registrant information, including a *phone number*, *e-mail address* and *street address* . A real phone number is not free and can potentially be traced back to a perpetrator, but even providing a false phone number incurs a certain cost to generate lists of valid looking phone numbers, yet which do not carry any similarity. Most public e-mail providers have infrastructure in place to prevent automated account creation. Some registries perform an address check, to ensure street addresses are actually existing addresses, which forces perpetrators to provide existing addresses. Creating a new, counterfeit registrant account for each new domain registration is costly, as also registrars have *captchas* or similar measures in place to prevent automated account creation.

Therefore, Premadoma's use of *registrant data* as part of the detection strategy is an important instrument, both to increase the overall cost of sustaining large-scale malicious campaigns, as well as to improve the detection accuracy of malicious domain registrations and therefore lower the success rate of malicious registrations. Ultimately, Premadoma aims to reduce the sustainability of running large-scale campaigns.

1.1.2 Predicting malicious registrations. To establish this disincentivization, Premadoma applies machine learning techniques to predict at registration time whether or not the domain will be used in cybercriminal operations, based on similarities in registrant data and the reuse of facilitators.

By deploying Premadoma as part of a registry infrastructure, predicted domains are prevented from being added to the registry's zone file, and the cybercriminal operations are preemptively rendered harmless.

1.2 Contributions

Firstly, we propose and develop Premadoma, an operational solution for DNS registries to predict malicious campaign domains at registration time. The system uses a novel machine learning-based strategy employing two complementary predictive models: a reputation-based classification (Section 3) and a similarity-based clustering (Section 4), using a total of 38 features, 33 of which we believe are novel.

Secondly, we realistically evaluate the proposed prediction model's performance and adaptability by incorporating an 11-month testing phase. We thereby address complex and domain-specific challenges, such as coping with incomplete ground truth (Section 5).

Thirdly, we successfully deploy Premadoma in production for a top ccTLD registry. The system contributed to the take down of 58,966 registrations in 2018 (Section 6).[1]

2 SYSTEM OVERVIEW

The goal of Premadoma is to reduce the number of large-scale campaign registrations by increasing the cost for attackers. By operating in the production environment of a registry, we are able to prevent registrations from entering the zone file by predicting their maliciousness. In this section, we present the overall machine learning methodology that enables the system to make predictions about incoming registrations.

2.1 Machine learning methodology

2.1.1 Data collection. **Registry data.** We obtained 14 months of domain registration data from the *.eu* ccTLD registry, between April 1, 2015 and May 31, 2016. For each of the 824,121 new domain registrations, the following data fields were captured:

- *Registration fields:* domain name, registrar used, registration time and name server information.
- *Registrant contact fields:* (company) name, email address, phone and postal address.

Blacklist data. To determine whether or not a domain is used in malicious activity, three public blacklists are queried on a daily basis for 30 days, starting from the date of registration. We consult Spamhaus DBL [20], SURBL [19] and Google's Safe Browsing list [7]. As soon as a registration appears on one of these blacklists, we label it as *malicious*. In other words, a registration is labelled as *benign* by default and may receive an irreversible malicious label as time progresses.

Data enrichment. For each of the name servers listed during registration, the corresponding IP address is looked up, and where possible, matched to a geographical location [16]. Lastly, we precalculate some derivative features related to the domain name, including the lexical randomness score of the domain name, using the probability of character transitions [18] based on previously registered benign domains.

Prediction approach. We aim to detect malicious campaign registrations from two perspectives.

First, we focus on *facilitators of malicious registrations*: infrastructural and administrative services and components regularly used by cybercriminals (such as registrars and name servers). We

[1] This corresponds to about 1.5% of the entire registry's zone.

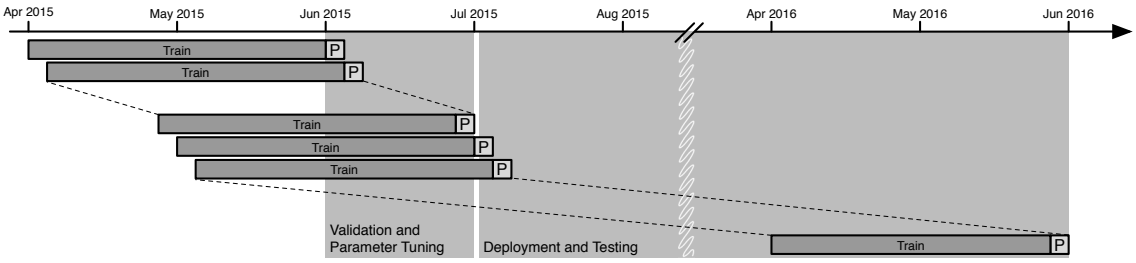

Figure 1: The validation period (June 2015) is used to select the best models and parameters. Afterwards, performance is tested on 11 months of registrations (July 2015 - May 2016). The prediction process uses a sliding window on a daily basis, i.e., each day, the model is trained on the previous 15 to 60 days.

Table 1: Featureset used for reputation-based classification (*'Cla'* culumn) and similarity-based clustering (*'Clu'* column).

Feature	Type	Cla	Clu	New?
domain_length	Ord	✓	✓	[9]
domain_digits	Ord	✓		[3]
domain_max_digit_len	Ord	✓		✓
domain_max_digit_offset	Ord	✓		✓
domain_max_hex_len	Ord	✓		✓
domain_max_hex_offset	Ord	✓		✓
hour_of_registration	Ord	✓		[9]
registrant_country_code	Categ.	✓	✓	✓
registrant_address_score	Cont.	✓	✓	✓
registrar_reputation_pct	Cont.	✓		✓
nameservers_reputation_pct	Cont.	✓		✓
email_provider_reputation_pct	Cont.	✓		✓
phone_number_reputation_pct	Cont.	✓		✓
registrar_reputation_pct_14d	Cont.	✓		✓
nameservers_reputation_pct_14d	Cont.	✓		✓
email_provider_reputation_pct_14d	Cont.	✓		✓
phone_number_reputation_pct_14d	Cont.	✓		✓
registrar_reputation_pct_30d	Cont.	✓		✓
nameservers_reputation_pct_30d	Cont.	✓		✓
email_provider_reputation_pct_30d	Cont.	✓		✓
phone_number_reputation_pct_30d	Cont.	✓		✓
registrar_reputation_pct_60d	Cont.	✓		✓
nameservers_reputation_pct_60d	Cont.	✓		✓
email_provider_reputation_pct_60d	Cont.	✓		✓
phone_number_reputation_pct_60d	Cont.	✓		✓
registrar	Categ.	✓	✓	[1, 5]
email_prov	Categ.	✓	✓	✓
registrant_street	String		✓	✓
registrant_phone	String		✓	✓
registrant_email_account	String		✓	✓
registrant_name	String		✓	✓
registrant_city	String		✓	✓
registrant_postcode	String		✓	✓
registrant_state_province	String		✓	✓
randomness	Cont.		✓	✓
seconds_since_last_reg	Categ.		✓	✓
nameservers_domains	Categ.		✓	[5]
nameservers_locations	Categ.		✓	✓

gauge the historical reputation of these facilitators and feed that information into a classifier. These *reputation-based predictions* are further described in Section 3.

Secondly, we try to detect whether a new registration is *part of a malicious campaign based on registrant data and facilitators*. To this end, we use unsupervised learning to group similar malicious domain registrations into clusters. Thereafter, we can assess whether a new registration is part of an existing malicious cluster. These *similarity-based predictions* are further described in Section 4.

Both detection strategies aim to be complementary: while the similarity-based prediction focuses on specific campaigns, the reputation-based predictions detect commonly used facilitators across campaigns. We therefore expect that ensemble learning techniques combining both prediction models can improve the strength of the overall prediction.

2.1.2 Daily training and datasets. To establish a truly autonomous and adaptive prediction system, we opt to retrain models on a daily basis, taking into account any new training data that becomes available. This sliding window ensures that evolving adversary tactics are continuously captured. To enable configuration of this aspect, we construct sliding training windows of different lengths (15, 30, 45 and 60 days).

Furthermore, as depicted in Figure 1, we split the datasets up into two phases:

Validation phase June 2015 is used for *selecting and tuning* the final prediction model and its parameters.

Testing phase During the testing phase (July 2015 - May 2016), the selected models are *evaluated* on unseen data. To ensure proper testing of resilience and robustness, the testing phase covers 11 months of registrations.

2.1.3 Evaluation criteria. We perform the evaluation of Premadoma from two perspectives. We use (1) the blacklist data (Section 2.1.1) as a basis for our ground truth, and (2) compare our results with the manual, post-factum analysis performed by Vissers et al. [22] on the same *.eu* TLD. The two primary evaluation metrics are precision and recall.

The *recall* or *true positive rate* (TPR), is the percentage of all blacklisted domains that the model was able to correctly predict as malicious.

$$recall = TPR = \frac{TP}{TP + FN}$$

The *precision* or *positive predictive value* (PPV) expresses from all registrations that were predicted to be malicious, how many were truly so.

$$precision = PPV = \frac{TP}{TP + FP}$$

Due to the overwhelming majority of benign registrations, the dataset is highly unbalanced, making precision a more adequate metric to compare the performance of the different models than, for instance, the false positive rate (FPR).

In the next two sections, the two predictive models of PREMADOMA are introduced, and evaluated in terms of precision and recall on the validation set.

3 REPUTATION-BASED PREDICTION

The main goal of the reputation-based predictor is to identify and track facilitators of campaign registrations, and use their past reputation as an indicator of maliciousness. The selection of a suitable classifier will be based on its prediction performance as well as on the human interpretability of the classification model, which is an operational requirement for registry customer support teams to be able to handle possible false positives.

3.1 Reputation of the facilitators

To register domain names in bulk or in long-running campaigns, malicious actors need different facilitators to enable these registrations. These include the **registrar** through which the registrations are made and the **name servers** used to set up the DNS resolution for the domain name. Other facilitators are communication means, such as the **email address** and **phone number**. These are for instance used by the registrar to comply with the ICANN Whois accuracy program [12].

For the reputation-based prediction, *reputation score* features have been engineered for these four prominent facilitators. The reputation scores express the percentage of registrations linked to a particular registrant's phone number, registrar, e-mail provider or name server, which were labelled as malicious in the ground truth data. These reputation scores are calculated daily over 4 different time windows: *14*, *30*, *60 days*, as well as an *all-time* window that takes into account all available historical data.

This feature set is combined with 6 features derived from the domain name, 1 ordinal feature (hour of registration), 1 continuous feature (address validity score) and 3 categorical features (country, registrar and email provider), as shown in Table 1. [2]

3.2 Classification algorithm

A good combination of prediction performance and interpretability was reached using the *PART* algorithm, proposed by Frank and Witten [6]. This algorithm uses the training data to iteratively build an ordered list of '*if-then*' rules as prediction model by constructing partial C4.5 decision trees in each iteration and using the 'best' leaf of the tree as a new rule. The resulting model is an ordered list of rules, combining registration feature inequalities by means of multiple AND-clauses.

[2]The features in Table 1 with a checkmark in the *cla* column are features used for classification.

The resulting model is human interpretable and closely resembles the configuration of (handcrafted or curated) rule-based systems, yet since it is generated by a machine learning algorithm, it automatically adapts to changing adversarial techniques, therefore combining the best of two worlds.

3.3 Addressing dataset challenges

The dataset used in this research suffers from some inherent limitations. First, the classes are highly imbalanced: only a small minority of registrations is malicious. Second, not all malicious registrations are correctly flagged in the training sets, due to delays and incompleteness of the ground truth. We address these challenges to ensure our system performs in a real-world context.

In the following paragraphs, we propose a number of methods to counter these inherent dataset challenges.

Blacklisting delays. According to [22], it takes up to 5 days for 73% of malicious registrations to be flagged by blacklists. As a result of this delay, each day the training set contains malicious domains that haven not yet been flagged. This can negatively skew the prediction models.

To absorb this effect, we filter out recent benign registrations from the training set, as these might still turn out malicious in the near future. In our current approach, we filter out the benign registrations from the last 5 days.

Class imbalances. The data set contains many more benign than malicious domains: With just 2.53% of registrations labelled as malicious in the validation and testing phase, we face an extremely unbalanced dataset. Without proper precautions, this situation results in a high bias towards predicting registrations as benign.

We analyze the effects of this imbalance by applying class subsampling, a common machine learning technique. This enables us to configure a preferred ratio of benign and malicious class instances in the training data. We name this parameter the *distribution spread*.

Incomplete ground truth.

Unfortunately, some malicious domains will never appear on a blacklist. According to [22], 19% of domains registered as part of malicious campaigns remain undetected.

This incompleteness of the ground truth makes it challenging to train consistent models. To mitigate this issue, we improve the consistency of the training set by removing registrations with potential anomalous labelling from the training set. In case the fraction of a registrant's domains above a configurable threshold is flagged as malicious, benign registrations from the same registrant are considered as potentially missed and therefore removed from the training set.[3] We name this threshold parameter the *blacklist incompleteness (bli)*.

For instance, a *bli* score of 100% means that no anomalies are pruned, whereas a *bli* score of 80% prunes benign registrations from a registrant which has more than 80% malicious registrations in the past.

At the same time we take a non-optimistic approach for ground truth incompleteness in terms of evaluation. Specifically, we do not remove these blacklist anomalies while evaluating the predictions. This strict measure negatively impacts prediction performance

[3]For simplicity, the registrant's phone number is used here as a proxy for the registrant.

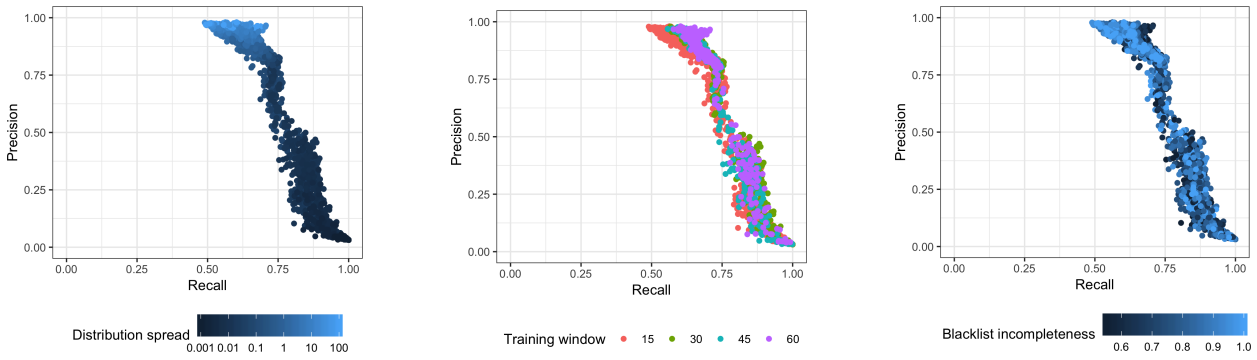

Figure 2: Performance comparison of the different reputation-based prediction models in terms of precision and recall during the validation phase.

metrics, but is a necessary vantage point for handling real-world registrations. As a result, some of the false positives reported in this paper might be read as true positives, which were unfortunately not flagged by the consulted blacklists.

3.4 Parameter tuning

We look at the following parameters to determine the best reputation-based model configuration in the validation phase.

Distribution spread	0.001 - 100
Blacklist incompleteness (bli)	55-100%
Training window	15, 30, 45 and 60 days

For each day in the validation phase (Figure 1), we execute a training and prediction step to determine the overall performance of a configuration. Figure 2 shows the performance of each setting in terms of precision and recall. As depicted, the different configurations achieve a trade-off between precision and recall.

As shown in Figure 2a, a large **distribution spread** preserves a class imbalance with a majority of registrations being benign, and as a result optimizes towards a high precision/low recall trade-off. Lowering the distribution spread makes malicious registrations more prominent in the training set, and steadily increases the recall at the cost of precision.

The 15 day **training window** achieves a lower recall than training sets of a longer period, especially for predictors with a high precision (Figure 2b). This illustrates that the classifier needs sufficient samples to correctly identify and generalize patterns.

As expected, a low threshold for **blacklist incompleteness** compensates for the ground truth incompleteness, and achieves a better recall. This is particularly noticeable in Figure 2c for predictors with a precision above 85%.

4 SIMILARITY-BASED PREDICTION

In this section, we propose a prediction method to autonomously cluster malicious registrations by leveraging the perceived similarities that these malicious registrations share. These clusters of malicious registrations are then used to predict whether new instances are associated with ongoing malicious activity, i.e. campaigns.

The proposed system operates as follows, as illustrated in Figure 3. First, benign registrations are discarded, while similar blacklisted registrations are clustered together with the aim of representing "campaigns". The goal is to obtain a small set of dense clusters of associated malicious registrations. Next, for each new registration, the pair-wise distance between the new registration and the blacklisted registrations is assessed. In case the new registration belongs distance-wise to one of the malicious clusters, the new registration is predicted as malicious.

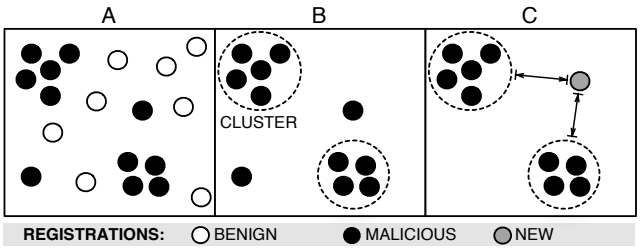

Figure 3: The three phases of the similarity-based prediction process. Malicious registrations in the training set are grouped together in clusters. Afterwards, new registrations can be compared to those clusters to predict their association with malicious activities.

The remainder of this section discusses in more detail the similarity metric and the clustering technique used in this prediction approach.

4.1 Similarity metric between registrations

To cluster blacklisted registrations, we need the ability to assess the similarity of two instances. We propose a custom similarity metric that expresses the distance between two registrations. This metric is then used in the clustering and prediction phases.

Similarity features. The distance metric primarily takes into account *registrant data*, alongside the domain name, name servers and registrar used during the registration. These features are either numerical, categorical or string-based:

String features (Company) name, address, postal code, city, state / province, email and phone of the registrant

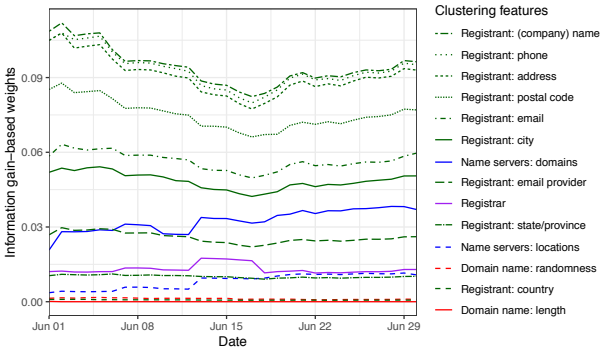

Figure 4: Daily variation of information gain-based feature weights for the similarity-based predictions (year 2015)

Numerical features Length, randomness score of the domain name (as introduced in Section 2.1.1), address validity score.

Categorical features Email provider and country of the registrant, domains used for name servers and their GeoIP location (country), and the registrar

For string-based features, the similarity is expressed as a normalized Levenshtein distance [25]. This distance between two strings is defined as the minimum number of single-character adjustments (insertions, deletions or substitutions) that are needed to transform one string into the other. The distance function thus preserves a notion of partial similarity and common characters in both strings. We opt for a similarity metric rather than equality/inequality to cope with typos and small variations as well as mitigation for possible evasion. Since registrations vary in length, a normalized pairwise distance is used.

The distance between numerical features is expressed as the Euclidian distance of the values. To account for differences in range, the numeric features are first normalized using a min-max-scaler across the entire dataset of malicious registrations before their distances are calculated.

Categorical features have a limited set of possible values. The similarity distance of categorical features is therefore expressed by equality (0 or 1). As name server geolocations can have multiple values, the similarity distance for these is expressed as the fraction of values that are shared between the two instances.

Pairwise distance metric. The pairwise feature distance of two registrations i and j is expressed as f_{ij}. The total pairwise distance between the two registrations is a weighted sum of the pairwise feature distances.

$$d_{ij} = d_{ji} = \sum_f w_f \cdot f_{ij}$$

Information gain-based weights. We aim to make the distance metric resilient to changing adversary tactics by enabling autonomous feature re-weighting. To define the relative importance of the various features in the pairwise distance, we calculate the information gain of each individual feature via multi-interval discretization [4]. The information gain expresses to what extent a feature can partition the registrations in benign and malicious registrations.

$$w_f = IG\left(\begin{array}{c} \text{Malicious reg.} \mid \text{Benign reg.} \\ \left[\begin{array}{ccc|ccc} f_{11} & \cdots & f_{1n} & \cdots & f_{1m+n} \\ f_{21} & \cdots & f_{2n} & \cdots & f_{2m+n} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ f_{n1} & \cdots & f_{nn} & \cdots & f_{nm+n} \end{array}\right] \end{array} \right)$$

To calculate the information gain, pairwise distances need to be calculated (1) between malicious registrations and (2) between malicious and benign registrations.

Figure 4 plots the daily information gain of the features during the validation phase. Clearly, the registrant features (in green) and the name server domains (in blue) are the most prominent.

Although the absolute feature weights vary on a day-by-day basis in Figure 4, their relative importance is quite stable over time. Therefore, the clustering weights could be reused for multiple days, as further discussed in Section 6.2.

4.2 Clustering algorithm

We opt for Agglomerative Clustering, given its ability to work with custom pairwise distances. Agglomerative clustering belongs to the family of hierarchical clustering algorithms and works by iteratively merging two clusters that are the closest to each other [11].

In order to merge the most similar clusters, the algorithm must be able to determine the distance between clusters. For this purpose, we adopt the complete linkage criterion. Using this criterion, the distance between two clusters is equal to that of the most dissimilar instances of both clusters, promoting a high intra-cluster similarity.

Distance matrix. The distance matrix DM is a symmetric $n \times n$ matrix, where n is the number of malicious registrations in the training set, and is constructed from the custom pairwise distances d_{ij}. This distance matrix DM serves as the input for the agglomerative clustering algorithm.

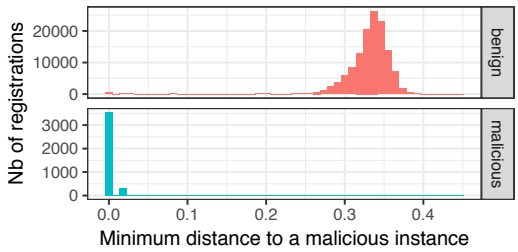

Figure 6: Distribution of the minimal distance to a malicious registration.

Clustering distance threshold. To partition into disjoint clusters, a stop criterion needs to be provided to the hierarchical clustering algorithm. As a cutting point, we supply a maximum distance threshold. Clusters are then merged until the maximum distance is reached. Similarly, this distance threshold is used to assess whether or not a new registration belongs to an existing cluster of malicious registrations.

To determine an optimal value for this clustering threshold, we plot the distribution of the smallest pairwise distance found to any

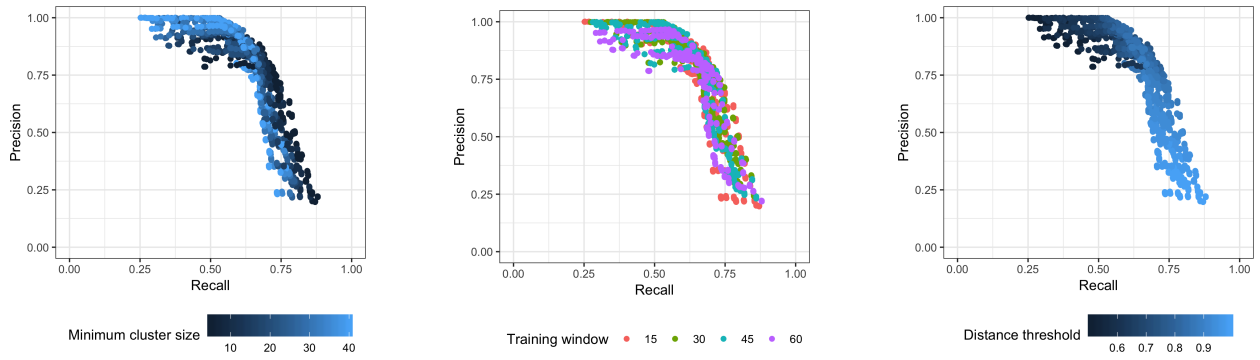

Figure 5: Performance comparison of different similarity-based prediction models in terms of precision and recall during the validation phase.

malicious registration, for both the malicious and benign domains in the training set (Figure 6).

As expected, malicious registrations are in general really similar to another malicious registration (a consequence of the campaign ecosystem). In contrast, the very vast majority of benign registrations are substantially dissimilar from malicious registrations. As such, we can express the clustering distance threshold as a value between the average of the malicious and benign minimal pairwise distances. This *distance threshold parameter* is expressed as a normalized value between those two averages.

Minimum cluster sizes. Finally, we introduce a threshold on the minimum number of registrations within a cluster. As such, only clusters with a sufficient number of malicious registrations during the training phase are taken into account for predicting the maliciousness of a new registration. Setting a higher *minimum cluster size* shifts focus towards larger, long running campaigns, whereas a smaller minimum cluster size picks up newly started campaigns more quickly.

4.3 Parameter tuning

We look at the following parameters to determine the best similarity-based model configuration in the validation phase.

Distance threshold	0.50 - 1.00
Minimum cluster size	5 - 50 registrations
Training window	15, 30, 45 and 60 days

Similar to Section 3.4, we execute a training and prediction step for each day in the validation phase (Figure 1) to determine the overall performance of a configuration. Figure 5 shows the performance of each setting in terms of precision and recall. As depicted, the different configurations achieve a trade-off between precision and recall.

A noticeable trend in Figure 5c is the impact of the **distance threshold**. A lower threshold achieves a higher precision as only registration very similar to malicious instances are withheld, while a higher threshold tends towards a higher recall.

Reducing the **minimum cluster size** pushes the predictor towards a higher recall, as new small campaigns are also taken into account (Figure 5a). In contrast, with a high minimum cluster size, the model only predicts registrations that are part of the largest campaigns.

Counterintuitively, the 60 day **training window** is not able to achieve the near 100% precision, as is the case with the other training windows (Figure 5b). However, this is not due to the training window size itself, but rather due to our static range for minimum cluster sizes: for a training window of 15 days, a cluster size of 30 registrations is a big factor, whereas this is less the case for a 60 day training window. To counter this, a bigger range of cluster sizes could be tested.

5 FINAL PREDICTIVE MODEL

In this section, ensemble predictors are designed by combining the models of the reputation-based classification (Section 3) and similarity-based clustering (Section 4). During the validation phase (shown in Figure 1), the best performing ensemble is selected. Afterwards, we move to the testing phase and present a in-depth evaluation of the selected ensemble on the 11-month testing set.

5.1 Majority voting model

For the two complementary prediction strategies, we have already evaluated the performance of their different configurations on the validation set. Now, we apply majority voting [14] to construct ensembles of any three base predictors coming out of both Section 3 and 4. The results of these ensembles in the validation phase are displayed in Figure 7. For clearer visualization, we limit the plot to the envelopes of best-performing reputation-based and similarity-based predictors, next to the best-performing majority voting ensembles. The ensemble models clearly achieve better results in both precision and recall.

In terms of model selection, we choose the ensemble with the highest F_1-score. The F-score expresses the weighted harmonic mean of precision and recall, enabling evaluation of a model's performance through a single metric.

$$F_\beta = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

The selected ensemble is annotated in Figure 7. It combines the following base models in a majority vote:

	Predictor type	Training	Parameters
P_1	Similarity-based	45 days	Minimum cluster size: 40 Distance threshold: 0.80
P_2	Reputation-based	15 days	Distribution spread: 0.006 Blacklist incompleteness: 80%
P_3	Reputation-based	60 days	Distribution spread: 20 Blacklist incompleteness: 60%

Table 2: Majority voting ensemble with the highest F_1 score

It should be noted that the choice of the most appropriate predictor model depends on the particular business use case, and is a trade-off between precision and recall. For some use cases (such as pre-emptive blocking domain registrations), a high precision (and low false-positive rate) is indispensable, for other use cases (proactive monitoring) a higher recall might be more suitable. To make such selections, other F-scores (e.g. $F_{0.5}$ or F_2) can be used to put more emphasis on either precision or recall.

5.2 Testing the ensemble model

We evaluate the ensemble model from the perspective of blacklist data, as well as by using campaign knowledge.

Ground truth-based evaluation. In order to evaluate the performance of the selected ensemble model on unseen data, we run the daily retrained model on the 11-month testing phase and compare its predictions with the ground truth labels.

In this testing phase, the ensemble model achieves 66.23% recall at a precision of 84.57% and a false positive rate of 0.30%.

Campaign-based evaluation. In addition to evaluating the predictions with respect to the ground truth, we evaluate how well the ensemble model is able to predict long-running campaigns. For this purpose, we compare the model's prediction results with the post-factum manual campaign analysis by Vissers et al. [22] over the same set of domains registrations.

17 out of the 20 campaigns are well predicted, leading to an overall recall of 76.68% and a precision of 87.20% with respect to campaign registrations. The 3 campaigns that were more difficult for Premadoma to predict were campaigns c_05, c_12 and c_15, which employ some of the evasion patterns, described in Section 7.1:

Campaign c_05 is a small campaign with a blacklist coverage below 60%. Moreover, the registrants use a WHOIS privacy protection service to hide registrant details. This practice violates terms and conditions of this registry, and as such already forms a reason for immediate suspension.

Campaign c_15 is a prototypical example of an advanced campaign, as discussed in detail by Vissers et al. [22]: the campaign applies 98 different registrant contacts, typically varies on a daily basis and only 27% of the domains end up being flagged by blacklist services.

Campaign c_12 Similarly, campaign c_12 is composed of 29 different registrants, all being used on at most 2 distinct days.

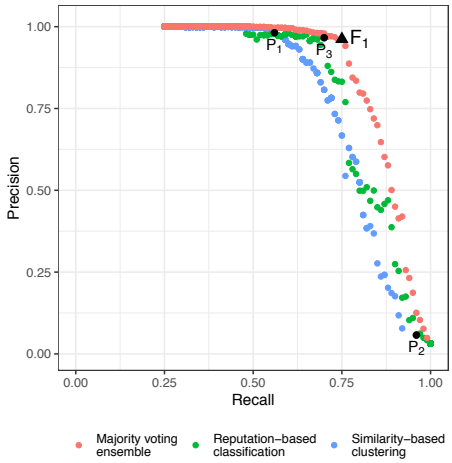

Figure 7: Performance comparison of the ensemble models in terms of precision and recall during the validation phase. The F_1 model is selected for use in the testing phase.

6 REAL-WORLD DEPLOYMENT

In this section, we report on how we placed Premadoma in production at the registry, detailing on the operational challenges and results.

6.1 Phases of the deployment

The models proposed in sections 3 to 5 have been successfully deployed at the registry in several phases.

Incremental development. Early 2017, the integration in the TLD's operational infrastructure was prepared and fine-tuned. From mid 2017 onwards, the system processed incoming real-world registrations on a daily basis.

Real-time detection and manual prevention. Since 2018, the system was deployed at the operational environment of the registry. Training was performed on a daily basis and predictions were made in real-time. The detection results were fed to the business and legal teams for follow-up. A central point of evaluation in the operational environment was the stability of the prototype, as well as the speed and the accuracy of the predictions.

Real-time detection and automatic prevention. The system is currently shifting from detection to automatic prevention: In 2019, gradually more and more incoming registrations will be automatically delayed from entering the registry's zone file, based on the predictions of the Premadoma system, and domain owners will be contacted for identity verification.

6.2 Accuracy and performance trade-offs

To run Premadoma in an online business context, we had to make several trade-offs between performance and prediction accuracy. An important driver hereby is that, in contrast to typical research experiments, the computer resources to run the system are limited and that there is a need for (near) real-time decisions about new incoming domain registrations. To that extent, several performance

tactics have been applied to speed up the training and prediction of the similarity-based model.

Cached Levenshtein distances. The calculation of normalized Levenshtein distances for pairwise string similarity (Section 4.1) is computationally intensive: on average, more than 1 billion pairwise distances need to be calculated to quantify the information-gain based weights during a daily training.

As a performance tactic, the pairwise feature distances are cached between training days, and as such only about 5% of normalized Levenshtein distances need to be calculated each day without affecting the prediction accuracy.

Static feature weights. The information-based weights w_f (Section 4.1) require pairwise feature distances between benign and malicious training instances (Section 4.2). On average, a training window consists of 1,805 malicious domains and 83,094 benign domains.

By using static weights only malicious-to-malicious feature distances need to be calculated, and resource-intensive information-gain calculation can be omitted. We determine these static weights by averaging the dynamic weights in the validation period. This performance tactic reduces the training effort with a factor of 40.

A downside of using static weights is the reduced adaptability to changes in the campaign registration ecosystem. This can be compensated by updating the static weights at discrete moments in time.

Clustering distance threshold. To optimize the speed of prediction, the complete linkage criterion is chosen. Hereby, the distance threshold represents the maximum distance within a single cluster. As a result, for a new domain registration only a minimal number of pairwise distances need to be calculated: typically only n pair-wise distances for benign registrations and $n + m$ distances for malicious registrations.[4] This optimization significantly speeds up the prediction time.

6.3 Feedback for business and legal teams

PREMADOMA provides a human interpretable context to the business and legal teams for each malicious prediction. This helps them to understand the underlying reason for which an incoming domain name is predicted as malicious.

For domains predicted by the similarity-based model, we highlight the features of the registrations that are most similar to those of the malicious domains in the matching cluster. Additionally, we list the domains that make up that cluster. Similarly, for the reputation-based model, insights derived from the rules of the decision list offer a human-interpretable insight in the model and are internally reported to the relevant teams.

6.4 Operational results

Figure 8 displays the prediction performance of the deployed ensemble model from July 2017 to December 2018. The red area plots the total number of registrations that were eventually blacklisted on a week-by-week basis, whereas the green area represents the successful predictions. Notice, that the number of malicious domain registrations at the *.eu* TLD dropped significantly in the second half of 2018, illustrating the claim stated in section 1, that malicious actors actively adapt their strategies.

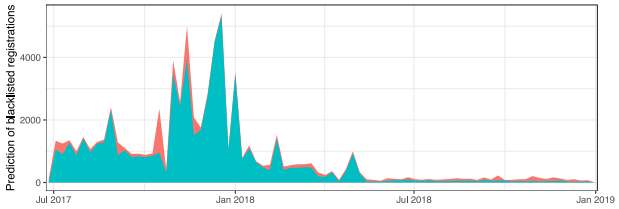

Figure 8: The weekly prediction of blacklisted registrations. The green area plots correct predictions, the red area are not-predicted blacklisted domains. Note the effect the deployment had in 2018 on the number of malicious registrations.

Suspension of abusive domains. As noted in Section 6.1, PREMADOMA is deployed as part of the registry's security program since January 2018. As a result of this program, 58,966 domains were suspended in 2018, which corresponds to about 1.5% of the total TLD zone.

Performance statistics. PREMADOMA is currently deployed at the registry's site on a single virtual machine with 2 Intel® Xeon® CPUs running at 2.50GHz and 24 GB RAM. It predicts incoming domain registrations on average in 0.52 seconds, with the 95% percentile at 0.97 seconds, which satifies the requirements of the *.eu* ccTLD. If required, the system can trivially be scaled out to perform predictions in parallel.

The training time depends on the number of malicious registrations in the training set, and can take up to 1 hour in a single-threaded setting.

7 DISCUSSION

In this section, we discuss the impact of evasion and abuse attempts against PREMADOMA, followed by an overview of potential limitations of this research.

7.1 Evasion patterns

Despite the adaptive training strategy, attackers may still attempt to circumvent the detection by exploiting specific details of the training mechanism. In our work, we start from the observation that malicious actors register domains in large quantities and do so in characteristic patterns. PREMADOMA assumes that a daily trained model can learn on early registrations of these domains, and predict subsequent registrations. We have identified three evasion patterns that try to undermine this basic assumption, and propose appropriate mitigation strategies.

Day-to-day variance. To predict campaign registrations, PREMADOMA requires similarities with malicious registrations from the training set or their facilitators. Therefore, the first instances of a completely new campaign, e.g. using new facilitators and registrant details, are not detected. Malicious actors can leverage this by devising unpredictable registration techniques every day. This

[4] Here n stands for the total number of malicious clusters in the training set, whereas m represents the number of malicious registrations in the matching cluster. In the non-optimized scenario $n \cdot m$ pair-wise distances need to be calculated for each incoming domain registration.

approach has already been observed in practice, e.g. campaign c_15, as reported in [22].

However, the cost for attackers increases to achieve this level of circumvention. Moreover, it makes campaigns less efficient and less scalable, ultimately fulfilling the goal of the system to deter campaigns through economic means. Additionally, our daily training strategy can be easily adapted to update the model more frequently (e.g. hourly or continuously) if necessary.

Periods of inactivity. Training sets have a limited view of 15 to 60 days prior to prediction. To avoid detection, an attacker can register a batch of domain names on one day, and wait a sufficient amount of time before registering their next batch.

Once more, this evasion strategy significantly decreases the efficiency and scalability of campaigns. In addition, the training window of the prevention system can be enlarged if required. Moreover, the reputation-based predictor is not entirely limited to the training window, as it takes into account historical reputation as well.

Overshoot in registered domains. Given the impact of ground truth incompleteness, attackers may register more domains than needed as part of their strategy. As such, they positively influence their reputation scores in order to remain undetected. Inherently, this evasion tactic again increases the cost for attackers, as it requires registering domains that cannot be deployed in malicious operations. Moreover, this issue is already partially addressed by introducing blacklist incompleteness parameters (Section 3.3).

All three evasion tactics are realistic threats and to some extent already deployed in practice by the more advanced campaigns. However, they all incur higher costs, cooldown or setup effort per successful campaign registration and thus satisfy our goal to economically disincentivize large campaigns.

7.2 Abuse of the system itself

Given the automated approach of Premadoma, malicious actors could also deploy adversarial attacks. In particular, though not trivial, a denial-of-service attack could be achieved if a perpetrator were to introduce a large amount of new malicious registrations in the system that closely mimic benign registrants and registration facilitators. As such, both the reputation-based and similarity-based models would start linking these patterns to malicious behavior and block registrations made by a bonafide entity.

Although such an adversarial attack is hard to prevent in the DNS registration ecosystem, it requires a substantial amount of injected malicious registrations, which, due to the fact that they must closely mimic benign registrations, comes at an even higher cost for the attacker. Moreover, we propose an appeal procedure, similar to the ones currently in place with blacklists, in which a bonafide domain owner can contest erroneous decisions.

7.3 Limitations

Here, we discuss four limitations of this study and the proposed predictive model.

Firstly, our proposed solution relies on blacklists for seeding the daily training sets. Unfortunately, blacklists are inherently incomplete and, while we attempt to limit the impact of that, the quality of our predictions depends on the quality and availability of the ground truth labels.

Secondly, several registrars offer WHOIS privacy services to their customers, obscuring the registrants contact information. Evidently, this diminishes the ability to differentiate between registrations of the same registrar and conceals information that Premadoma relies upon. In the case of the *.eu* ccTLD, the use of WHOIS privacy services actually violates the registry's terms and conditions, and are in itself a reason for suspension. [5]

Additionally, Premadoma is focused on registrations belonging to large-scale malicious campaigns. There is a residual minority of malicious registrations, appearing as "one-offs", that are not explicitly targeted by the system.

Lastly, the goal of Premadoma is to increase the cost for attackers to register malicious campaign domains. It is however hard to quantify the exact cost increase necessary to bypass the system we propose. Each campaign will have a different economic model and return on investment, making it hard to gauge at which cost attackers will be deterred. Nonetheless, by placing Premadoma in production and conducting several takedowns, we witness a strong reduction in the amount of malicious registrations in 2018 (see Figure 8).

8 RELATED WORK

In previous work, Vissers et al. [22] extensively analysed 14 months of registration data to identify large-scale malicious campaigns present in the .eu TLD. Their insights in the ecosystem of malicious domain registrations directly underpin the foundations of the detection and prevention techniques proposed in this paper.

Prior to our research, Hao et al. [10] studied the domain registration behavior of spammers and suggested that, given the use of large-scale campaigns, registries and registrars are well-positioned to interfere with bulk registrations by malicious actors. The effects of several registrar-level interventions had already been documented by Liu et al. [15]. A key concern the latter authors raise, is the ability of attackers to quickly and easily change to a different, non-intervening registrar.

In a more recent paper, Hao et al. [9], introduce PREDATOR, a system that can be used by registrars or registries to detect malicious domain registrations at registration time. PREDATOR is an important predecessor of this research and has evaluated the feasibility of registration-time detection. In contrast, Premadoma was designed and evaluated in the operational setting of a top ccTLD. Our proposed method therefore strongly focused on tactics to handle the inherent domain-specific challenges such as ground truth imbalance and blacklist incompleteness. PREDATOR and Premadoma operate with very different feature sets (as shown in Table 1) and make use of very different classifiers. According to a survey of detection techniques by Kidmose et al. [13], only very few detection methods use registrant data (neither does PREDATOR). Yet, as can be seen from Figure 4, the registrant-related features (in green), actually exhibit the highest information gain.

Unfortunately, though, at this time, a fair performance comparison between PREDATOR and Premadoma is not feasible, given

[5] The WHOIS privacy service for GDPR compliance that is active for individuals who register *.eu* domain names is implemented by the *.eu* TLD itself.

that they were evaluated on very different data sets and on different TLDs. This would constitute interesting future work.

In line with the goal of our work, several other studies aim to reduce the time necessary to stop abuse by identifying related malicious domains. Felegyhzi et al. [5] investigated the feasibility of proactive domain blacklisting, by inferring other malicious registrations from known-bad domains through shared name servers and identical registration times. PREMADOMA leverages this concept of facilitators in a more generic way as part of the reputation-based classification, and applies this approach to registration-time prevention.

Many studies concentrate on DNS traffic of newly registered domains to detect malicious behavior [1–3, 8, 17, 23, 24]. For instance, Xu et al. [24] use passive DNS data to predict malicious behavior, and Weber et al. [23] evaluated a number of different clustering techniques for identifying malicious domain campaigns. These systems generally focus on the operational DNS patterns of domain names, while PREMADOMA engages in an earlier stage, before the domain is in use. A recent study by Vissers et al. [21] revealed that approximately 20% of domain registrations by malicious actors are not picked up by blacklists, stressing the importance of alternatives to blacklisting, as well as explaining the sometimes higher than expected False Positive Rates, as a significant fraction of False Positives are actually True Positives that did not end up on blacklists.

9 CONCLUSION

In this paper, we presented PREMADOMA, a system for DNS registries to predict malicious domain registrations at registration time. This system takes a novel approach by simultaneously focusing on recurring patterns in registrant information as well as targeting commonly used registration facilitators. Using this approach, PREMADOMA substantially lowers the success rate of malicious registrations and facilitator reuse, and hinders the sustainability of running large-scale malicious campaigns. We thoroughly evaluated the proposed prediction model's performance and adaptability on an 11 month testing set, and addressed complex and domain-specific dataset challenges, such as coping with incomplete coverage of blacklists. Since the successful deployment at the *.eu* ccTLD registry, PREMADOMA already contributed to the takedown of 58,966 malicious registrations and has lead to a remarkable decline in malicious domain registrations.

REFERENCES

[1] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee, and Nick Feamster. Building a dynamic reputation system for dns. In *Proceedings of the 19th USENIX Conference on Security*, pages 18–18, 2010.

[2] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, II, and David Dagon. Detecting malware domains at the upper dns hierarchy. In *Proceedings of the 20th USENIX Conference on Security*, pages 27–27.

[3] Leyla Bilge, Sevil Sen, Davide Balzarotti, Engin Kirda, and Christopher Kruegel. Exposure: a passive dns analysis service to detect and report malicious domains. *ACM Transactions on Information and System Security (TISSEC)*, 16(4):14, 2014.

[4] Usama M. Fayyad and Keki B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*, pages 1022–1029, 1993.

[5] Mark Felegyhazi, Christian Kreibich, and Vern Paxson. On the potential of proactive domain blacklisting. In *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, pages 6–6, 2010.

[6] Eibe Frank and Ian H. Witten. Generating accurate rule sets without global optimization. pages 144–151. Morgan Kaufmann, 1998.

[7] Google. Google Safe Browsing, 2016. https://developers.google.com/safe-browsing/.

[8] Shuang Hao, Nick Feamster, and Ramakant Pandrangi. Monitoring the initial dns behavior of malicious domains. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 269–278. ACM, 2011.

[9] Shuang Hao, Alex Kantchelian, Brad Miller, Vern Paxson, and Nick Feamster. Predator: Proactive recognition and elimination of domain abuse at time-of-registration. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 1568–1579, New York, NY, USA, 2016. ACM.

[10] Shuang Hao, Matthew Thomas, Vern Paxson, Nick Feamster, Christian Kreibich, Chris Grier, and Scott Hollenbeck. Understanding the domain registration behavior of spammers. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, pages 63–76, 2013.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. 2001.

[12] ICANN. 2013 Registrar Accreditation Agreement, 2013. https://www.icann.org/resources/pages/approved-with-specs-2013-09-17-en#whois-accuracy.

[13] Egon Kidmose, Erwin Lansing, Søren Brandbyge, and Jens Myrup Pedersen. Detection of malicious and abusive domain names. In *Data Intelligence and Security (ICDIS), 2018 1st International Conference on*, pages 49–56. IEEE, 2018.

[14] Louisa Lam and SY Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 27(5):553–568, 1997.

[15] He Liu, Kirill Levchenko, Márk Félegyházi, Christian Kreibich, Gregor Maier, Geoffrey M. Voelker, and Stefan Savage. On the effects of registrar-level intervention. In *Proceedings of the 4th USENIX Conference on Large-scale Exploits and Emergent Threats*, 2011.

[16] MaxMind, Inc. GeoLite2 Free Downloadable Databases, 2016. https://dev.maxmind.com/geoip/geoip2/geolite2/.

[17] Giovane CM Moura, Moritz Müller, Maarten Wullink, and Cristian Hesselman. ndews: A new domains early warning system for tlds. In *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, pages 1061–1066. IEEE, 2016.

[18] Rob Renaud. Gibberish Detector. https://github.com/rrenaud/Gibberish-Detector.

[19] SURBL. SURBL - URI Reputation Data, 2016. http://www.surbl.org.

[20] The Spamhaus Project Ltd. The Domain Block List, 2016. https://www.spamhaus.org/dbl/.

[21] Thomas Vissers, Peter Janssen, Wouter Joosen, and Lieven Desmet. Assessing the effectiveness of domain blacklisting against malicious dns registrations. In *4th International Workshop on Traffic Measurements for Cybersecurity (WTMC 2019)*, 2019.

[22] Thomas Vissers, Jan Spooren, Pieter Agten, Dirk Jumpertz, Peter Janssen, Marc Van Wesemael, Frank Piessens, Wouter Joosen, and Lieven Desmet. Exploring the ecosystem of malicious domain registrations in the. eu tld. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 472–493. Springer, 2017.

[23] Michael Weber, Jun Wang, and Yuchen Zhou. Unsupervised clustering for identification of malicious domain campaigns. In *Proceedings of the First Workshop on Radical and Experiential Security*, pages 33–39. ACM, 2018.

[24] Wei Xu, Kyle Sanders, and Yanxin Zhang. We know it before you do: predicting malicious domains. In *Proc. of the 2014 Virus Bulletin Intl. Conf*, pages 73–77, 2014.

[25] L. Yujian and L. Bo. A normalized levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1091–1095, June 2007.