FANCI : Feature-based Automated NXDomain Classification and Intelligence

Samuel Schüppen, *RWTH Aachen University;* Dominik Teubert, *Siemens CERT;*
Patrick Herrmann and Ulrike Meyer, *RWTH Aachen University*

https://www.usenix.org/conference/usenixsecurity18/presentation/schuppen

This paper is included in the Proceedings of the
27th USENIX Security Symposium.

August 15–17, 2018 • Baltimore, MD, USA

ISBN 978-1-931971-46-1

Open access to the Proceedings of the
27th USENIX Security Symposium
is sponsored by USENIX.

FANCI : Feature-based Automated NXDomain Classification and Intelligence

Samuel Schüppen
RWTH Aachen University

Dominik Teubert
Siemens CERT

Patrick Herrmann
RWTH Aachen University

Ulrike Meyer
RWTH Aachen University

Abstract

FANCI is a novel system for detecting infections with domain generation algorithm (DGA) based malware by monitoring non-existent domain (NXD) responses in DNS traffic. It relies on machine-learning based classification of NXDs (i.e., domain names included in negative DNS responses), into DGA-related and benign NXDs. The features for classification are extracted exclusively from the individual NXD that is to be classified. We evaluate the system on malicious data generated by 59 DGAs from the DGArchive, data recorded in a large university's campus network, and data recorded on the internal network of a large company. We show that the system yields a very high classification accuracy at a low false positive rate, generalizes very well, and is able to identify previously unknown DGAs.

1 Introduction

Modern botnets rely on domain generation algorithms (DGAs) for establishing a connection with their command & control (C2) server instead of using fixed domain names or fixed IP addresses [14, 2]. According to DGArchive[1], to date more than 72 different DGAs are known and the number is expected to further increase [14] as DGAs significantly improve a botnet's resistance against takedown. A DGA generates a set of malicious algorithmically-generated domains (mAGDs) serving as potential rendezvous domains with a C2 server. The bots subsequently query the domain name system (DNS) for the IP addresses of these domains. The amount of domains generated per day varies between 1 and 10,000 depending on the DGA [14]. The botmaster registers a few of these mAGDs. If these are queried by the bots, the bots obtain a valid IP address for their C2 server. All of the many other queries of the bots will result in non-existent domain (NXD) responses.

In the past, monitoring DNS traffic (successfully resolving and/or non-resolving) has been used as primary or additional source of information in detecting malicious activity in a network (e.g., [2, 16, 18, 9, 4]). Some of these approaches have concentrated on identifying C2 servers, others have focused on identifying infected devices or detecting malicious domains in general. These prior approaches, however, all require the correlation of information extracted from groups of DNS queries and/or responses and thus typically require extensive tracking. In addition, many of these prior approaches are based on clustering, which involves manual labelling of the identified clusters. While these prior works show promising detection capabilities, little information on the efficiency of the detection process in terms of time and memory requirements is reported.

This work presents FANCI: Feature-based Automated NXDomain Classification and Intelligence, a novel system for detecting infections with DGA-based malware by monitoring NXD responses. FANCI's classification module uses an machine learning (ML)-classifier (random forests (RFs) or support vector machines (SVMs)) to separate NXDs into benign non-existent domains (bNXDs) and mAGDs. This classifier uses a small number of language-independent features that can efficiently be extracted from the domain names included in NXD responses alone. Other contextual information extracted from the full NXD response that carried the domain name, from other related DNS responses, or from any other source are not required.

We extensively evaluate FANCI's classification module on malicious data obtained from DGArchive [14] and data recorded in the campus network of RWTH Aachen University[2], and in the internal network of the Siemens AG[3]. The evaluation shows that FANCI is able to detect unknown DGAs with a detection accuracy of over 99% at a very low false positive rate. Unlike prior work,

[1] https://dgarchive.caad.fkie.fraunhofer.de/

[2] https://www.rwth-aachen.de

[3] https://www.siemens.com

we also show that FANCI generalizes very well, that is, it maintains its detection quality even when applied to data recorded in a network different from the one it was trained in. Applying FANCI, we were able to identify ten DGAs not included in the DGArchive at the time of writing. We reckon that at least four of them were completely unknown, while the others most likely result from unknown seeds or are variations of known DGAs. Finally, our system is very efficient with respect to both training (5.66 min on 92,102 samples) and prediction (0.0025 s per sample) such that it is even able to perform on-the-fly detection in large networks without sampling.

FANCI's lightweight feature design and its generalizability allows for versatile application scenarios, including the use of its classification *as a service*, and its use in large-scale networks as well as on home-grade hardware.

```
bknllsnbfzqr.net
cdzogoexis.tv
hdozpcy.com
```

(a) Kraken

```
3lgrupwdivsfm2w4kng2iha.ddns.net
ojyvips6klsnqpy.in
af5fmb78sbuno4c.ws
```

(b) Corebot

```
salt-amount-pattern.com
company-depend.com
btkindasaladmw.com
```

(c) Matsnu

```
getadobeflashplayer.net
egtadobeflashplayer.net
etadobgeflashplayer.net
```

(d) VolatileCedar

```
rbtqebf.biz
qaskebf.com
qaskebf.biz
```

(e) Torpig

```
kea174638023becce522b1ae8f6caadf80.to
l8743f7debd036e5de923bbd70a191d009.in
ma4dbf2b2ef5bb0d01a065198fab552b25.hk
```

(f) Dyre

Figure 1: Illustration of mAGDs of six different DGAs.

```
univresity.edu
iieee.org
mcirosfot.com
adobe.comm
```

(a) Typing error

```
wfnfhde
kaqoeizerbo
ahxurofbdughh.rwth-aachen.de
pphrncxkxe.itsec.rwth-aachen.de
```

(b) Google Chrome

```
brn001ba99bbcd9.matha.rwth-aachen.de
cache-cdn.kalaydo.com
fileserverfb6.fb6.rwth-aachen.de
de-swyx-2.fraba.local
```

(c) Misconfiguration

Figure 2: Illustration of typical bNXDs from the network of RWTH Aachen University.

2 Preliminaries

In this section, we provide a brief overview on the types of mAGDs different DGAs generate and categorize different types of domain names that occur in NXD responses due to benign causes. This is followed by an overview of the supervised learning classifiers we use in this work. Note that throughout this work, we always use *NXD response* to refer to the entire UDP[4] packet containing the DNS response. In contrast, we refer to *NXD* as the bare domain name included in such a response.

2.1 Domain Names in NXD Responses

In order to highlight the diversity in the generation schemes used by different DGAs, Figure 1 illustrates example mAGDs of six different DGAs. Where mAGDs generated by *Kraken*, *Corebot*, and *Torpig* look completely random, the mAGDs of *Matsnu* are concatenations of genuine English words. mAGDs of *VolatileCedar* are all permutations of the same base domain name and *Dyre* generates mAGDs of equal length that consist of a 3 character prefix followed by a hash-like string.

In addition to NXDs generated by DGAs (i.e., mAGDs), there are mainly three groups of benign non-existent domains (bNXDs) originating from typing errors, misconfigurations, and misuse, respectively, where misconfiguration and misuse belong to the group of benign algorithmically-generated domains (bAGDs). bAGDs are, like mAGDs, generated algorithmically but originate from benign software and only have benign purposes. Typing error bNXDs are caused by humans misspelling existing domain names. Misconfiguration bAGDs are caused by devices or software trying to resolve domain names that do not exist (anymore) due to configuration errors or bugs. Misuse bAGDs are typically caused by software using DNS for non-intended purposes. For example, anti-virus software performing signature checks [17] or Google Chrome, which uses random domain names to probe its DNS environment and detect DNS hijacking attempts [19]. Figure 2 shows example bNXDs for each of the three categories.

2.2 Supervised Learning Classifier

In our work, we focus on *supervised learning classifiers*, more specifically on random forests (RFs) and support vector machines (SVMs) using the two labels *benign* and *malicious*. The labels are known for training purposes.

An RF is an ensemble of multiple decision trees (DTs) introduced to overcome limitations of a single DT. Predicting the label of an unknown sample using an RF is performed by a majority vote of all DTs in the forest. RFs were originally introduced in [10] and later on refined, for example, in [5, 6].

An SVM computes a hyperplane during training to

[4] in rare cases TCP is used

separate the training data according to their label. Then, unknown data can be predicted by determining the location of an observed sample in relation to this hyperplane. SVMs were introduced by Vapnik [7].

3 Features

In this section, we describe the 21 features used by FANCI to classify NXDs into bNXDs and mAGDs. We divide the presented features into three categories: structural features, linguistic features, and statistical features. We focus on features that are computationally lightweight w.r.t. their extraction, do neither require precomputations, nor a priori knowledge, and are independent of a specific natural language.

Our feature design is naturally inspired by the features used in related work [14, 2, 16]. However, we focus on features that can be extracted from an individual domain name. In particular, we get rid of all features used in previous work that require additional contextual information without loss of (in fact rather increasing) accuracy (see Section 6).

3.1 Definitions and Notation

Throughout the rest of this paper we use the notations detailed in the following.

A *domain name d* is a sequence of characters from an alphabet Σ. It consists of a sequence of subdomains separated by dots: $d = s_n. \dots .s_2.s_1$, where $s_i, i \in \{1,\dots,n\}$ denotes the i-th subdomain of d. Note that the permitted alphabet Σ in legitimate domain names depends on local registration authorities. Theoretically, almost all Unicode characters are permissible [13].

A *valid top level domain (TLD)* is a TLD that is part of the official list of TLDs maintained by the Internet Assigned Numbers Authority (IANA), for example, `org`, `com`, `eu`, and `edu` [3]. Currently, 1,547 valid TLDs are listed in the root zone [11].

A *public suffix* is a suffix under which domains are publicly registrable. This includes valid TLDs as well as suffixes, such as `dyndns.org` or `co.uk`. The Mozilla Foundation maintains a list of more than 11,000 valid public suffixes[5] [8].

A *feature* is defined as a function $\mathscr{F}$of a sample d, where $\mathscr{F}(d)$ denotes the *extracted feature*. $\mathscr{F}(d)$ can either be a single scalar or a vector of scalars. Concatenating all extracted features results in the *feature vector* of d. In the following sections, some of our features (marked by *) ignore separating dots and some (marked by †) ignore valid public suffixes. Features ignoring both operate on a string referred to as *dot-free public-suffix-free domain* and denoted by d_{dsf}. Consider for example the domain name $d =$ `itsec.rwth-aachen.de` that yields $d_{dsf} =$ `itsecrwth-aachen`.

#		Feature	Output	$\mathscr{F}(d_1)$	$\mathscr{F}(d_2)$
1		Domain Name Length	integer	19	34
2	†	Number of Subdomains	integer	2	2
3	†	Subdomain Length Mean	rational	7.5	25
4		Has `www` Prefix	binary	0	0
5		Has Valid TLD	binary	1	1
6	†	Contains Single-Character Subdomain	binary	0	0
7		Is Exclusive Prefix Repetition	binary	0	0
8	†	Contains TLD as Subdomain	binary	0	0
9	†	Ratio of Digit-Exclusive Subdomains	rational	0.0	0.0
10	†	Ratio of Hexadecimal-Exclusive Subdomains	rational	0.0	0.0
11	*†	Underscore Ratio	rational	0.0	0.0
12	†	Contains IP Address	binary	0	0

Table 1: Illustration of 12 structural features evaluated on the example domains d_1 and d_2, where $d_1 =$ `bnxd.rwth-aachen.de` and $d_2 =$ `dekh1her76avy0qnelivijwd1.ddns.net`. Some features (marked by *) ignore separating dots and some (marked by †) ignore valid public suffixes.

Note that we ignore separating dots in some of our features, because the *number of subdomains* feature already reflects the number of subdomains of a domain name and the dots as such do not provide any additional information. We ignore public suffixes in some features as they are not algorithmically generated. Although a DGA may vary the public suffix among its mAGDs, it is only able to choose from the official pool of available public suffixes as otherwise the resulting domain names would not be resolvable on the public Internet. As benign domain names have to select public suffixes from the exact same pool of officially available public suffixes, a public suffix offers no valuable additional information to distinguish mAGDs from bNXDs.

3.2 Structural Features

The first feature category focuses on structural properties of a domain name. Table 1 gives an overview of our structural features including an example evaluation on the domain names $d_1 =$`bnxd.rwth-aachen.de` and $d_2 =$`dekh1her76avy0qnelivijwd1.ddns.net`, where d_1 is benign and d_2 is a known mAGD.

In the following, we discuss the non-self-explanatory structural features #7, #9, #10, and #12 in more detail.

(#7) Is Exclusive Prefix Repetition. This is a binary feature, which is 1 if and only if the domain consists of a single character sequence w that

[5] `https://publicsuffix.org`

#		Feature	Output	$\mathscr{F}(d_1)$	$\mathscr{F}(d_2)$
13	†	Contains Digits	binary	0	1
14	*†	Vowel Ratio	rational	0.21	0.3
15	*†	Digit Ratio	rational	0.0	0.2
16	*†	Alphabet Cardinality	integer	12	18
17	*†	Ratio of Repeated Characters	rational	0.25	0.33
18	*†	Ratio of Consecutive Consonants	rational	0.67	0.36
19	*†	Ratio of Consecutive Digits	rational	0.0	0.08

Table 2: Overview of 7 linguistic features applied on the example domains d_1 and d_2.

is repeated at least twice. For example, for the domain name `rwth-aachen.derwth-aachen.de` this feature evaluates to 1, but for the domain name `rwthrwth-aachen.de` it evaluates to 0.

(#9) Ratio of Digit-Exclusive Subdomains. This feature is computed as the ration of the number of subdomains consisting exclusively of digits to the overall number of subdomains. It ignores public suffixes. Consider for example the domain name `123.itsec.rwth-aachen.de` resulting in $1/3$ as it has 3 subdomains (the public suffix `de` is excluded), where one of them consists of digits exclusively.

(#10) Ratio of Hexadecimal-Exclusive Subdomains. This feature is defined analogously to feature (#9) Ratio of Digit-Exclusive Subdomains.

(#12) Contains IP Address. This is a binary feature, which is 1 if and only if the domain contains an IP address, where IP address refers to common notations of IPv4 and IPv6 addresses including dots.

3.3 Linguistic Features

To extend our feature set we focus on linguistic characteristics of domain names in the following. These features are used to capture deviations from common linguistic patterns of domain names. Table 2 presents an overview of all 7 linguistic features. In the following, we discuss the non-self-explanatory linguistic features #17, #18, and #19 in detail.

(#17) Ratio of Repeated Characters. The *repeated character ratio* is computed on the d_{dsf} and is defined as the number of characters occurring more than once in d_{dsf} divided by the alphabet cardinality (#16). Considering the example domain name $d =$ `bnxd.rwth-aachen.de` this feature evaluates to $3/12$, where repeating characters in d_{dsf} are `n`, `h`, and `a`.

#		Feature	Output	$\mathscr{F}(d_1)$	$\mathscr{F}(d_2)$
20	*†	N-Gram Dist.	vector		
	1-Gram	d_1	$(0.43, 1, 1.25, 1, 2, 1, 1.25)$		
	1-Gram	d_2	$(0.59, 1, 1.39, 1, 3, 1, 2)$		
21	*†	Entropy	rational	3.64	4.05

Table 3: Overview of 2 statistical features evaluated on the example domains d_1 and d_2.

(#18) Ratio of Consecutive Consonants. This feature sums up the lengths of disjunct sequences of consonants ≥ 2 and divides the sum by the length of d_{dsf}. For example, considering the domain name $d =$ `bnxd.rwth-aachen.de` results in $(8+2)/15 = 0.67$, where $d_{dsf} =$ `bnxdrwth-aachen` and the consecutive disjunct consonant sequences are: `bnxdrwth` and `ch`.

(#19) Ratio of Consecutive Digits. This feature is defined analogously to feature (#18) Ratio of Consecutive Consonants.

3.4 Statistical Features

The two statistical features used by FANCI are shown in Table 3. Both are explained in detail in the following.

(#20) N-Gram Frequency Distribution [2]. An n-gram of domain name d is a multi set of all (also non-disjunct) character sequences e, $e \in d_{dsf}$, with $|e| = n$. f^n denotes the frequency distribution of the corresponding n-gram. The *n-gram frequency distribution* feature is defined as $g_n = (\overline{f^n}, \sigma(f^n), \min(f^n), \max(f^n), \tilde{f^n}, f^n_{0.25}, f^n_{0.75})$, where $\overline{f^n}$ is the arithmetic mean of f_n, $\sigma(f^n)$ the corresponding standard deviation, $\min(f^n)$ the minimum, $\max(f^n)$ the maximum, $\tilde{f^n}$ the median, $f^n_{0.25}$ the lower quartile, and $f^n_{0.75}$ the upper quartile. Table 3 exemplarily illustrates the evaluation of this feature for 1-grams on the domains d_1 and d_2. FANCI uses g_1, g_2, g_3 as feature #20 which results in a vector of 21 output values overall.

(#21) Entropy [14, 2]. The *entropy* (according to Shannon) is defined considering the 1-gram frequency distribution f^1 of d: $-\sum_{c \in d_{dsf}} p_c \cdot log_2(p_c)$, where p_c is the relative frequency of character c according to f^1. Table 3 shows example evaluations for the domains d_1 and d_2.

4 FANCI

In this section, we present *Feature-based Automated NXDomain Classification and Intelligence (FANCI).*

FANCI is a lightweight system for classifying arbitrary NXDs into benign and DGA-related solely based on domain names. It consists of three modules: training, classification, and intelligence. Figure 3 provides an overview of FANCI's architecture, of required inputs, of outputs, and of the way FANCI processes data internally. The three modules and potential application scenarios are described in more detail in the following.

4.1 Training Module

As FANCI is based on supervised learning classifiers, it requires training with labeled data. The training module implements training of classifiers and requires the input of labeled mAGDs and bNXDs (see upper left in Figure 3). We obtain labeled mAGDs for training purposes from DGArchive. Assuming FANCI operates in a campus or business network, bNXDs can for example be obtained from the network's DNS resolver. To obtain an as clean as possible set of bNXDs for training, we filter them in a *cleaning* step against all known mAGDs from DGArchive [14]. After the cleaning step, feature extraction is performed for each of the inputs as described in Section 3.

The output of the training module is a trained model, ready to be used for classification of unknown NXDs in the classification module.

4.2 Classification Module

The classification module classifies arbitrary NXDs into mAGDs and bNXDs based on a model it receives from the training module (see middle part of Figure 3). The classification module operates on an NXD, that is, on an individual domain name as input submitted for classification either by an intelligence module (see Section 4.3) or by any other source as indicated with a dashed arrow in Figure 3. The output of the classification module is a label for the submitted NXD that can take either of the two values *benign* or *malicious*.

To perform the classification, first, feature extraction is performed on the input NXD as described in Section 3. Afterwards, the actual classification is performed (currently either by RFs or by SVMs) on the extracted feature vector using the previously trained model. The classification module can either be used standalone or in combination with the intelligence module.

4.3 Intelligence Module

The intelligence module's task is to supply intelligence based on classification results, in particular, find infected devices and identify new DGAs or unknown seeds. As opposed to the classification module, which only takes the NXD itself as input, the intelligence module additionally takes the source and destination IP address and the timestamp of each NXD response as input in order to be able to map a malicious label as classification result back to the device that initiated the query.

In a first preprocessing step this module extracts the domain name and the aforementioned meta data from an NXD response. It uses the classification module to determine the label of the corresponding NXD and stores the results including the meta data in a database. To handle and improve results, postprocessing is performed, which can be divided into filtering and transformation.

Filtering is performed to further reduce false positives (FPs) and is carried out by filtering all positives against two whitelists. An NXD is removed from the positives list if it ends with a domain name present in one of the whitelists.

The first whitelist is of global nature and always applicable. It consists of the top X Alexa domains[6], where the exact amount X to use in this step is configurable. Whitelisting the top Alexa domains is based on the commonly made assumption that criminals are not able to host command & control (C2) servers under the most popular domains [4, 1]. To avoid whitelisting domain names such as `dyndns.org`, we exclude all domains from this list under which domains are publicly registrable according to Mozilla's list of public suffixes [8].

The second whitelist is of local nature. It considers domains occurring with high frequency in the network FANCI operates in. This list is fully configurable and we provide examples for two networks in the evaluation part of this paper (see Section 5.2.4).

After filtering, transformations are applied on the results to generate different views on this data and facilitate the analysis of the results. These transformations primary include the grouping of all positives by TLD or second-level domain, the grouping of NXDs by IP address of the requesting device, and the grouping by timestamps. Additionally, string-based searching and filtering of NXDs can be performed. Now, the data is well-prepared for a manual review and a conclusive interpretation.

4.4 Usage Scenarios

FANCI is a versatile and flexible system and is applicable in a variety of different scenarios. We mainly differentiate between two major use cases. The first case considers the usage of FANCI with all of its three modules at a single operation site, while the second case takes advantage of FANCI's modular design and considers a distributed use of FANCI.

[6] `https://www.alexa.com/`

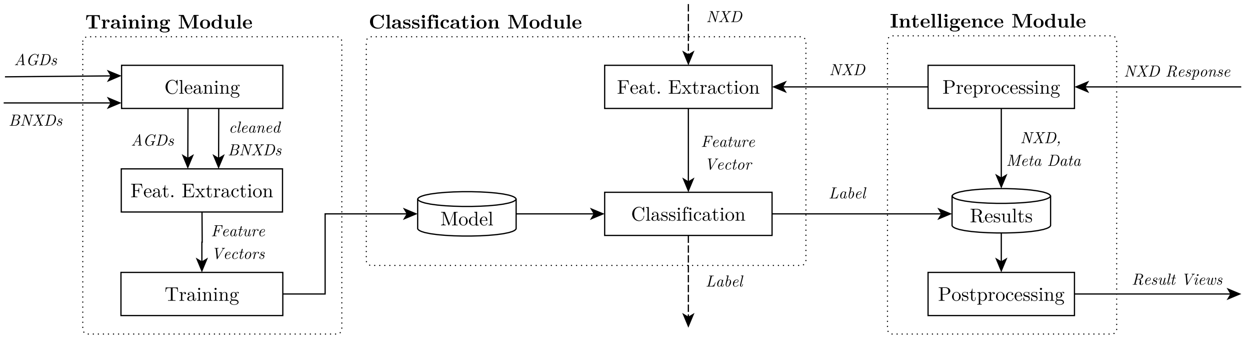

Figure 3: Abstract illustration of the architecture and operation of FANCI.

Local. This deployment scenario is typical for corporate or campus-grade networks, where FANCI can be used locally as a fully-featured system. Networks of this size usually have a centralized DNS infrastructure which eases the deployment of FANCI, in particular the acquisition of bNXDs to train the classifier and also subsequent real-time detection using NXD responses. In such a deployment the previously trained model is used to label NXDs and to provide insights about infected devices to network administrators and incident handlers.

In some networks (e.g., in a typical university network) DNS traffic of devices can be monitored in a way such that IP addresses of querying devices are visible. In this case, FANCI's intelligence module is able to map mAGDs detected in NXD responses to infected devices that queried them. The detection of an infected device may trigger a monitoring of the successfully resolved DNS traffic originating from these devices. Using FANCI's classification module trained on successfully resolved domains (see Section 5.5) then enables the detection of successfully resolving mAGDs and the identification of C2 servers allowing for blacklisting of corresponding IP addresses. Note that starting with monitoring the NXD responses only, has the advantage that much less traffic needs to be handled in this step than if we would monitor the full DNS traffic. As a DGA typically generates many more mAGDs that result in NXD responses than mAGDs that resolve, monitoring NXD responses is the most promising way to find infected devices. The chance an infected device is able to contact its C2 server before it has queried a non-resolving mAGD seem very slim.

In less permissive networks (e.g., in large corporate networks) DNS traffic may not allow for a direct mapping to devices, for example, because of a hierarchical DNS infrastructure, where central DNS servers only communicate with subordinate domain controllers. In this case, the identification of infected devices is less straight forward but could to some extend be managed with the help of sinkholing mAGDs detected by FANCI. FANCI could also be integrated into existing monitoring software and can significantly add value to its detection capabilities by providing directly utilizable threat intelligence. Domains that were classified as mAGDs by FANCI can be considered to be high-confidence indicators of compromise (IOCs). Thus, FANCI can trigger and support a variety of subsequent measures. This may include proxy log and DNS log analysis, for example to retrospectively detect further infections and to sinkhole or blacklist identified C2 domains. Furthermore, the utilization of detected mAGDs on host-based agents or network edge devices like routers or firewalls is possible to find further infected devices and disrupt C2 traffic at the same time.

Outsourced. FANCI generalizes well to unknown environments, which means that some parts can be outsourced. In particular, it is possible to perform training with data obtained from a certain campus-grade network and use the resulting model to perform detection in other networks. This enables the use of FANCI in networks, where it is hard to perform training. For example, this can be small networks (e.g., those of small businesses), where it takes too long to get the necessary amount of data for training or this can be networks, where it is a non-trivial task to obtain a clean set of bNXDs for supervised learning (e.g., ISP networks).

Furthermore, FANCI's classification module can be used *as a service*, for example, accessible via an API or a web service useable by security software or security researchers. Note that in this case, only the domain name in question would have to be submitted to the server. The entirety of labeled mAGDs could also further be shared using various mechanisms, for example, as a threat intelligence feed, which can again be integrated into existing protection efforts of large and medium-sized companies.

5 Evaluation

In this section, we present an extensive evaluation of FANCI's classification module. We compare SVMs and RFs to find the best performing classifier setup for detecting mAGDs and show that RFs slightly outperform SVMs in this use case. We show that FANCI's classification module generalizes well to unknown network environments and present a real world application test, whereby we are able to report new DGAs. Finally, we evaluated how well FANCI's classification module is able to detect resolving mAGDs in full DNS traffic. Before presenting our results in detail, we first describe our evaluation procedure, including a description of the data sets our evaluation is based on.

5.1 Data Sets

As FANCI's classification module relies on supervised learning classifiers, we require labeled data sets for training and evaluation. Furthermore, as classification is performed on domain names only, we only require sets of labeled unique domain names to evaluate classification performance. The three data sources we use are the RWTH Aachen University campus network, the internal network of Siemens AG and the DGA Archive [14].

RWTH Aachen University. The central DNS resolver of RWTH Aachen University serves as first source for bNXD responses, which includes a variety of academic institutes, eduroam[7], several administrative networks, student residences, and the University hospital of RWTH Aachen. The campus network is additionally interconnected with the University of Applied Science Aachen, and the Research Center Jülich [15]. Due to enforcement, a vast majority of devices uses the network's central DNS resolvers. Our bNXD data set is a continuous one-month recording of NXD responses recorded at the central DNS resolver. We recorded 31 days overall, more precisely from 22 May 2017 until 21 June 2017. In this one-month period, we recorded pcap files of NXD responses with a size of 98.9 GB containing approximately 700 million NXD responses, that is, on average we recorded 3.2 GB or 22.6 million NXD responses per day. In total, this data set comprises 35.8 million unique NXDs.

Siemens. As a second source for bNXDs we obtained data from the DNS infrastructure of Siemens. Note that we only obtained NXDs and not full NXD responses as this is entirely sufficient for FANCI's classification module. This data originates from several central DNS servers of Siemens AG and covers three regions: Europe, Asia, and the USA. This broad and international coverage guarantees diverse data from different entities and devices. We obtained data of a two-month period from September and October 2017 (i.e., 61 days) comprising 31.2 million unique NXDs overall.

The long recording periods for both benign data sets guarantee a representative data set including different times of the day, different days of the week, and working and non-working days. To clean our benign data sets as far as possible we checked our benign data against DGArchive [14] and removed all known mAGDs.

DGArchive. To obtain sets of known mAGDs we used the DGArchive [14]. mAGDs in DGArchive are computed by using reimplementations of reverse engineered DGAs and by using corresponding known seeds. Hence, DGArchive serves as an extremely reliable source for a malicious data set. Our data set comprises all data available from DGArchive at the time of writing. We were able to obtain mAGD data for 1,344 days, ranging from 12 February 2014 until 30 January 2018. In total, this set contains 72 different DGAs. As our selected ML algorithms at least need a set size of a few hundred NXDs to perform well, we decided to reduce the set by eliminating all DGAs with less than 250 unique mAGDs. This results in 59 remaining DGAs. For our malicious data set we consider unique mAGDs of these DGAs exclusively. This comprises 49,738,973 unique mAGDs in total. Across these DGAs, the number of unique mAGDs is between 251 and 13,488,000.

5.2 Classification Accuracy

In this section, we first determine the best performing classifier or ensemble of classifiers for detecting mAGDs. Next, we present several experiments, each to prove a certain capability of FANCI's classification module. This includes the ability to detect unknown seeds and unknown DGAs as well as showing that FANCI's classification module generalizes very well.

5.2.1 Experimental Setup

Due to the considerable size of our data set, we performed random sampling to generate sets for our evaluations. Each data set is composed of as many bNXDs as mAGDs, and is created by performing fresh uniform random sampling for each single set from our benign data sets. Depending on the corresponding experiment, the malicious data is either drawn uniformly at random

[7] Education Roaming—WLAN infrastructure for students and employees, `https://eduroam.org`

from the unique mAGDs of all DGAs or from the unique mAGDs of a single DGA. For sets considering all DGAs, we strive a uniform representation of all DGAs as far as possible. The size of a set here denotes the number of samples in total, that is, the sum of bNXDs and mAGDs.

Depending on the experiment we perform either a 5-fold cross validation (CV) or a leave-one-group-out (LOGO) CV. In a 5-fold CV the data set is divided into 5 equally sized folds using 4 for training and 1 for prediction. Each fold is used exactly once for prediction. Resulting statistical metrics are averaged over all 5 runs. An LOGO CV is in its basic procedure similar to a k-fold CV, but instead of building k random folds, the folds are defined regarding a predefined grouping, for example, by seeds or DGAs.

We determined the optimal parameter settings for the ML algorithms for two different scenarios with the help of extensive grid searches on data sets independent of the ones used for evaluation. The first scenario considers *single-DGA detection*, (i.e., one classifier targeting one specific DGA), where the second targets *multi-DGA detection* (i.e., one classifier trained to detect all DGAs). We fixed the resulting parameters and used them in all subsequent evaluation scenarios including the one done in the wild. For an excerpt of the results of the grid searches see Appendix B.

All computations were carried out on the RWTH Compute Cluster[8].

In all experiments, we consider accuracy (ACC) as primary metric to characterize a classifier's performance defined as $ACC = |TP| + |TN| / |population|$, where $|TP|$ is the amount of true positives and $|TN|$ the amount of true negatives. This means that ACC indicates the fraction of correctly predicted samples. However, for each experiment we additionally present statistics of the following four metrics: true positive rate (TPR), true negative rate (TNR), false negative rate (FNR), and false positive rate (FPR). For each metric we consider the arithmetic mean $\bar{x}$, the standard deviation σ, the minimum x_{min}, the median $\tilde{x}$, and the maximum x_{max}.

5.2.2 Classifier Selection

In this section, the presented experiments reflect the procedure to select the best performing classifiers for a real-world application. For the following experiments we consider benign data from RWTH Aachen exclusively. We performed each experiment for SVMs and RFs. As it is our goal to find the best performing classifier and RFs perform marginally better than SVMs in most scenarios, we present results for RFs in the following in detail. Results for SVMs can be found in Appendix A.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.99936	0.99989	0.99883	0.00011	0.00117
σ	0.00190	0.00050	0.00351	0.00050	0.00351
x_{min}	0.98600	0.99400	0.97267	0.00000	0.00000
$\tilde{x}$	0.99988	1.00000	0.99978	0.00000	0.00022
x_{max}	1.00000	1.00000	1.00000	0.00600	0.02733

Table 4: Results for classifying bNXDs and mAGDs of single DGAs with RFs. In total, 295 sets of 59 DGAs were considered each evaluated by 5 repetitions of a 5-fold CV.

Single DGAs. The first experiment covers the detection of a certain single DGA using a dedicated classifier. We considered all 59 DGAs and created 5 different sets per DGA of a maximum set size of 100,000 following the procedure presented in Section 5.2.1. This means that each data set always contains an equal number of mAGDs and bNXDs. Depending on the DGA less than 50,000 unique mAGDs may be available. In these cases the set size is adjusted accordingly. In summary, this yields 295 sets of a maximum size of 100,000. For each set we performed 5-fold CVs, which we repeated 5 times with fresh, random folds.

Table 4 presents a statistical description of an RF's capabilities in the detection of single DGAs. The mean ACC is 0.99936 with a small standard deviation of 0.00190. The minimal ACC of 0.98600 is reached in the detection of *Bobax*, which is the only outlier. RFs detect 6 out of 59 DGAs (*Bamital*, *Blackhole*, *Dyre*, *Sisron*, *Tofsee*, and *UD2*) with 100 percent ACC.

Unknown Seeds. In this experiment, we focus on evaluating the detection of mAGDs generated by a DGA with a new seed, where the model is trained with mAGDs generated by the same DGA using known seeds.

To evaluate this scenario we perform an LOGO CV, that is, we perform training with mAGDs of all but one seed of a certain single DGA, perform prediction on the skipped one, and repeat this procedure for each seed and DGA. Again, we use data sets with a maximum size of 100,000 and use 5 distinct sets per DGA. We consider all DGAs with at least two known seeds, which yields 30 DGAs with 550 seeds overall. In total, this results in $5 \cdot 550 = 2750$ iterations for all available seeds and DGAs.

A statistical summary of the evaluation results for this experiment for RFs is depicted in Table 5. The mean of the ACC is 0.95319 showing a notable standard deviation of 0.12499. ACC values are between 0.49900 and 1.0, where 75 percent of all measures show a higher ACC than 0.98193. As only 6 DGAs are related to an ACC lower than 98 percent, the wide range of the ACC can be

[8] `https://doc.itc.rwth-aachen.de/display/CC`

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.95319	0.90689	0.99947	0.09330	0.00053
σ	0.12499	0.25005	0.00075	0.25059	0.00075
x_{min}	0.49900	0.00000	0.99570	0.00000	0.00000
$\tilde{x}$	0.99965	0.99991	0.99960	0.00011	0.00040
x_{max}	1.00000	1.00000	1.00000	1.00000	0.00430

Table 5: Results for LOGO CV for mAGDs of single DGAs grouped by seed using RFs. In total, 150 sets of 30 DGAs were considered.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.99759	0.99764	0.99753	0.00236	0.00247
σ	0.00009	0.00013	0.00012	0.00013	0.00012
x_{min}	0.99745	0.99739	0.99733	0.00217	0.00228
$\tilde{x}$	0.99758	0.99762	0.99752	0.00238	0.00248
x_{max}	0.99776	0.99783	0.99772	0.00261	0.00267

Table 6: Results for detecting mAGDs with RFs of arbitrary mixed DGAs using 5 repetitions of 5-fold CV for each set. In total, 20 sets were considered.

explained by outliers.

This experiment is the only experiment, where SVMs perform slightly better than RFs. SVMs achieve a mean ACC of 0.98315 with a much smaller standard deviation of 0.06166, but with a similar wide range from 0.49850 to 1.0. Detailed results of this experiments for SVMs are presented in Table 14. SVMs are also affected by the same outliers (i.e., the same DGAs cause problems) as RFs. In contrast to RFs, SVMs do not consistently miss all new seeds of these certain DGAs and hence yield a slightly higher ACC in the mean.

Mixed DGAs. Next, we examine how well a single classifier trained on some mAGDs of the known DGAs is able to detect other mAGDs generated by one of these known DGAs.

We created 20 sets of a targeted size of 100,000 containing an equal number of mAGDs of each of the 59 DGAs. For DGAs with a too small amount (i.e., less than $50000/59 \approx 847$) of unique mAGDs we included all available mAGDs of such DGAs, which results in an effective set size of 92,102. For each of these 20 sets we performed 5 repetitions of a 5-fold CV.

In its trend, results for detecting mAGDs in sets containing mAGDs of multiple DGAs are similar to the detection of using dedicated classifiers for each single DGAs as presented previously. Table 6 illustrates measurement results for RFs. The ACC's mean is 0.99759 with a very small standard deviation of 0.00009. Minimum and maximum ACC values are 0.99745 and 0.99776 respectively.

In summary, we state a single classifier trained with mAGDs of multiple DGAs achieves a very high and stable ACC in detecting arbitrary mAGDs.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.98073	0.96389	0.99756	0.02424	0.00244
σ	0.00034	0.00065	0.00015	0.00072	0.00015
x_{min}	0.97972	0.96182	0.99726	0.02339	0.00221
$\tilde{x}$	0.98078	0.96397	0.99759	0.02416	0.00241
x_{max}	0.98119	0.96468	0.99779	0.02649	0.00274

Table 7: Results for LOGO CV for sets of mAGDs of mixed DGAs grouped by DGA using RFs. In total, 20 sets were considered.

Unknown DGAs. This experiment confirms capabilities in detecting mAGDs of unknown DGAs. To verify that our classifiers are able to generalize to mAGDs of unknown DGAs we performed LOGO CV regarding a grouping by DGA, that is, mAGDs of all but one DGA are used for training and mAGDs of the left out DGA are predicted. Sets considered in this experiment are equivalent to sets from the previous experiment, that is, we consider 20 sets with equal numbers of mAGDs per DGA. This means that for each of the 20 sets we performed 59 iterations of training and prediction leaving one DGA out at once.

Table 7 depicts a statistical summary of results for RFs in detecting mAGDs of unknown DGAs. The ACC is between 0.97972 and 0.98119 and the mean of the ACC is 0.98073 with a very small standard deviation of 0.00034. RFs detect 55 out of 59 left out DGAs with an ACC comparable to the previously presented experiment. We conclude that we are able to detect mAGDs of unknown DGAs.

Classifier Selection. In real-world applications, we aim at reliably detecting known DGAs as well as unknown seeds and DGAs. Furthermore, we want to achieve maximum classification accuracy. Hence, we have to choose the best performing classifier or ensemble of classifiers to achieve these goals. For this reason, we additionally evaluated several logical combinations of classifiers dedicated to single DGAs. In particular, we tested several *or* and *and* combinations, threshold voting with different thresholds, majority voting, even with combinations of RFs and SVMs. However, a single RF classifier trained with all known DGAs outperforms any of the above ensembles. That is why FANCI uses a single RF classifier trained with mAGDs of all known DGAs.

5.2.3 Generalization

Up to now, we performed all experiments with test sets containing bNXDs from RWTH Aachen University. In

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.99699	0.99815	0.99582	0.00185	0.00418
σ	0.00015	0.00018	0.00022	0.00018	0.00022
x_{min}	0.99681	0.99787	0.99540	0.00132	0.00372
$\tilde{x}$	0.99697	0.99812	0.99581	0.00188	0.00419
x_{max}	0.99730	0.99868	0.99628	0.00213	0.00460

Table 8: Results for classifying mAGDs of arbitrary mixed DGAs and bNXD from Siemens applying 5 repetitions of 5-fold CV for 20 sets each of size 100,000 using RFs.

this section, first, we show that FANCI performs with the same quality when trained and deployed in a different network. Second, we demonstrate that it is even possible to perform training with data recorded in one network and use the resulting classification model in another network. This means that FANCI generalizes well to new environments.

Mixed DGAs; Training and Prediction Siemens. To illustrate FANCI's detection capabilities are independent of a certain network, we repeated the *mixed DGA experiment* from Section 5.2.2 but with sets generated with bNXDs from the Siemens data set. This experiment yields ACC values comparable to those obtained in the same setting for RWTH data. The mean ACC is 0.99699 with a small standard deviation of 0.00015, where the minimum is 0.99681 and the maximum is 0.99730. Table 8 illustrates the detailed detection performance when using data from the Siemens network.

Next, we carry out two experiments proving that our trained classifiers generalize well to unknown networks, that is, we examine the scenario of training a classifier using data from a certain network but use this classifier somewhere else. To evaluate our loss in ACC when using a classifier trained in a foreign network we compare the ACC to scenarios, in which we trained and predicted using bNXDs from the same network.

Mixed DGAs, Training RWTH, Prediction Siemens The first experiment considers training using bNXD from RWTH Aachen and performs prediction on sets composed with bNXDs from Siemens. The second experiment is performed vice versa. These experiments are based on the fact that mAGDs do not differ from network to network, but only bNXDs may be different. For both benign data sources we consider 20 data sets each generated as in the previous experiments. Each data set is used for training once, where prediction is performed for each of the 20 sets of the other bNXD source. This results in $20 \cdot 20 = 400$ passes for each of the two experiments.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.99534	0.99937	0.99132	0.00063	0.00868
σ	0.00018	0.00007	0.00034	0.00007	0.00034
x_{min}	0.99511	0.99920	0.99083	0.00051	0.00799
$\tilde{x}$	0.99530	0.99939	0.99125	0.00061	0.00875
x_{max}	0.99565	0.99949	0.99201	0.00080	0.00917

Table 9: Classification accuracy for training on RWTH Aachen data and prediction on Siemens data using RFs.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.99785	0.99946	0.99624	0.00054	0.00376
σ	0.00009	0.00006	0.00019	0.00006	0.00019
x_{min}	0.99771	0.99936	0.99591	0.00048	0.00349
$\tilde{x}$	0.99784	0.99946	0.99622	0.00054	0.00378
x_{max}	0.99800	0.99952	0.99651	0.00064	0.00409

Table 10: Classification accuracy for training on Siemens data and prediction on RWTH Aachen data using RFs.

Table 9 presents results for considering sets containing bNXDs from RWTH Aachen for training and sets containing bNXDs from Siemens for prediction. The mean ACC is 0.99534, with a small standard deviation of 0.00018. In comparison to performing training and prediction on sets containing bNXDs from Siemens (see Table 8), the mean ACC is only marginally smaller, namely 0.00165 percentage points. This is explained by an increase of FPs. However, the false negatives (FNs) even decrease.

Mixed DGAs, Training Siemens, Prediction RWTH Table 10 shows results for considering sets containing bNXDs from Siemens for training and bNXD data from RWTH Aachen for prediction. In this experiment the mean ACC is 0.99785, which is in comparison to the RWTH-only (see Table 6) experiment even marginally larger, namely by 0.00026 percentage points. Although the FPs increase slightly, the FNs decrease. This confirms the trend from the previous experiment.

Again, we performed all experiments with SVMs and RFs and RFs perform consistently better than SVMs. Results for SVMs can be found in Appendix A. In summary, the previous experiments show that FANCI is in general independent of a certain network, generalizes well to unknown environments, and even allows for outsourcing of the actual classification.

5.2.4 Additional False Positive Reduction

As highlighted in Section 4.3, FANCI performs a filtering in the intelligence module to reduce FPs. To evaluate the efficiency of our filtering approach we consider sets

Initial	Alexa top *X*	Alexa		Alexa + Local	
		red. by %	rem.	red. by %	rem.
RWTH 6,522	10^2	0.08	6,517	75.53	1,596
	10^4	71.79	1,840	77.69	1,455
	10^6	86.49	881	89.88	660
Siemens 11,431	10^2	0.31	11,395	47.85	5,961
	10^4	7.52	10,571	53.12	5,359
	10^6	74.18	2,952	77.74	2,544

Table 11: False positive reduction applied with and without local specific whitelist, where the reduction is presented in percent (red. by %) and the remaining amount of FPs (rem.) is additionally stated as absolute value.

of all unique FP bNXDs occurred during experiments presented in the previous sections. As we use a local specific whitelist in the second filtering step, we consider two data sets, one for RWTH Aachen FP bNXDs (6,522) and one for Siemens FP bNXDs (11,431). We evaluated the global filtering step using the Alexa top 100, top 10,000, or top 1,000,000. The local specific filtering is performed with appropriate whitelists for each of the networks. For the RWTH Aachen University network, this list for example includes domains, such as, `rwth-aachen.de`, `sophosxl.net`, and `fh-aachen.de`. For the Siemens network, this list for example contains: `siemens.net`, `trendmicro.com`, `mcafee.com`, and `bayer.com`. These local specific whitelists assume that there is no C2 server present in the campus networks. Additionally, we assume that certain companies, such as, Sophos, McAfee, and TrendMicro do not host a C2 server.

Table 11 presents the results of applying both filtering steps subsequently on these two sets of unique FP bNXDs. It states the reduction of FPs in percent and the amount of remaining FPs. For data from RWTH Aachen we are able to reduce the FPs by 75.53 up to 89.88 percent, which results in 1,596 or 660 remaining FPs respectively. Considering the Siemens network, we reduce the FPs at least by 47.85 percent resulting in 5,961 domains and in the best case we reduce the FPs by 77.74 percent yielding 2,544 domains left.

The results clearly show the efficiency of our subsequent FP filtering. Although FANCI's classification accuracy is already outstanding, we are able to at least halve the amount of FPs even when only considering the Alexa top 100 as whitelist. In the best case we are even able to reduce FPs to a tenth of the initial amount.

Now, that we have seen FANCI's capabilities in detecting mAGDs and proved efficiency of our false positive reduction we present a real world application of FANCI in the next section.

5.3 Real World

In this section, we present the application of FANCI in the university network of RWTH Aachen.

Setup. For our real world application test of FANCI we consider a fresh one-month recording from the central DNS resolver of RWTH Aachen University comprising 31 days, more precisely from 13 October 2017 until 12 November 2017, where the data amount is similar to the recording from Section 5.1. This means that FANCI has to handle approximately 700 million NXD responses in total, containing 35 million unique NXDs. FANCI is used with a single RF classifier trained on a set of size 92,102 containing mAGDs of 59 different DGAs and bNXD from RWTH Aachen network from the data set described in Section 5.1. The set contains bNXDs and mAGDs in equal parts and equal many mAGDs of each DGA. We applied FANCI by first using the classification module on all NXD responses from the fresh recording and then used the filtering capabilities of the intelligence module for FP reduction using Alexa's top 1,000,000.

Results. Applying these two steps we obtained 22,755 unique positive NXDs ($\sim 0.065\%$) that occur in 45,510 NXD responses ($\sim 0.0065\%$) in total. After a semi-automatic examination of these remaining positives, we are able to report 405 unknown mAGDs corresponding to ten different groups either indicating an unknown DGA (UD) or an unknown seed (US). To find groups of unknown mAGDs we make use of the different views provided via FANCI's intelligence module as presented in Section 4.3. Note that unknown, here, means that the found mAGDs neither are listed in DGArchive nor could be found via other common sources at the time of writing. We will submit all findings to DGArchive. Figure 4 shows representatives of each of the ten groups including a label indicating if we reckon the group as UD, as US, or if both seems possible. We carried out the labeling of the groups with the help of DGArchive, domain knowledge, and manual research.

By implication, we have seen at most 22,345 unique FPs in our one-month, real-world test resulting in a worst-case FPR of approximately 0.00064. As it is hard to determine correct ground truth in a real-world application, this FPR is only of limited significance. For statements about the quality of FANCI's classification capabilities, it is more promising to analyze the potential FPs in more detail. The set of potential FPs is characterized by a high diversity among the NXDs. Figure 5 shows twelve potential FPs seen in our real-world evaluation. They can be classified into two groups: human-generated and machine-generated. Where human-generated NXDs usually exhibit natural language patterns or are very sim-

```
cxoriilg.host
dcveyroohhuz.host
ktnotybgnqrjvkq.host
ndptbhn.host
qbeweonxhzlflh.host
zwchzomnkersegz.host
```

(a) UD1

```
blwemxb.ga
yinnic.gq
fyrrzx.ml
fhvfbhq.tk
ihrslrk.cf
xlajbu.cf
```

(b) UD2

```
agng78sagdfdkjdtwa108.com
agng78sagdfdkjdtwa177.com
agng78sagdfdkjdtwa225.com
agng78sagdfdkjdtwa316.com
agng78sagdfdkjdtwa948.com
```

(c) UD3

```
brn001ba9933850.net
brn001ba99fa1c7.net
brw48e244240e9d.net
brwc0f8da79205c.net
brwc48e8fbdfa3e.net
```

(d) UD4

```
ageihehaioeoaiegj.es
rohgoruhgsorhugih.hu
siiifibiiegiiciib.in
iapghahpnpnapcipa.mobi
goiaegodbuebieibg.name
abvainvienvaiebai.info
```

(e) US1 *Locky*

```
1917f71a77.club
1a984212aa.club
2f949298a5.club
129aa1a6f7.space
1459f4a279.space
1a984212aa.space
```

(f) US2 *Infy*

```
539aa5d47547.com
646892faf047.com
52dd1bce8b10.net
646892faf010.net
853b3eb55b98.net
```

(g) US3 *PandaBanker*

```
3fdqrbnum3fa2j1.3tfrmn27i.com
c4xf33p7nrvo3l0h.23bjj3a0.com
wpdcp7uym0.up18xtxzouumzd.com
wlh8tj55fxfh.n51ah7y227y-.com
xgs66mu-uig2u.cjswb3q4m45.com
```

(h) UD5 or US4 of *Redyms*

```
afyonescortkizlar.xyz
ordubayanescort.xyz
kirikkalebayanescort.xyz
nigdebayanescort.xyz
bayanescortbandirma.xyz
bayanescortbilecik.xyz
afyonescortkizlar.xyz
```

(i) UD6 or US5 of *GozNym 2nd Stage / Nymaim*

```
getbeautifuljacked.xyz
evelynmiller.xyz
juicepress.xyz
quietbranch.xyz
tracyhernandez.xyz
webhostpremium.xyz
wertvollebrillanthobby.xyz
```

(j) UD7 or US6 of *GozNym 2nd Stage / Nymaim*

Figure 4: Illustration of unknown mAGDs.

ilar to existing domains, machine-generated NXDs tend to be either of random nature or of technical origin. Assigning an NXD to one of these classes is not always possible without additional information, for example consider the potential FP NXD `c.ssl-cd.com`, which could belong to each of the classes.

As there is no striking group of similar NXDs among the set of potential FPs, this allows us to conclude that FANCI makes no systematic classification errors underlining FANCI's extraordinary classification performance.

As the network of RWTH Aachen is secured by business security software and appliances using blacklists for known mAGDs, it is not surprising that we could find almost no known mAGD in our real-world test. To be precise, using DGArchive we were able to identify only 31 unique known mAGDs.

The application of FANCI in a month-month period in the university network of RWTH Aachen strikingly illustrates its detection capabilities in real world. Furthermore, this test emphasizes FANCI's ability to detect unknown mAGDs as well as known mAGDs. To further support FANCI's applicability in real, large-scale networks we present a consideration of FANCI's classification speed in the following.

```
eisenbahn-kurirer.de
rwth-aachend.de
www.cibc-global.hk
hotmail.om
www.digitex-eu.com
infonews24.org
```

(a) Human-Generated

```
fsztakqwdjfqsc.asa.at
isatap.host
ip38-201-hypermedia.net.id
103-56-7-42-mebd.net
1979775309.rsc.cdn77.org
host37-252.swifthighspeed.com
```

(b) Machine-Generated

Figure 5: Sample of potential FPs.

5.4 Training and Classification Speed

This section presents a brief overview of training and classification speeds to demonstrate FANCI's real-world applicability. All measurements were performed single-threaded on a Dell OptiPlex 980 with Intel i7 870@2.93GHz CPU and 16GB RAM running Ubuntu Linux 16.04. We performed training and classification 10 times for each of the mixed sets of size 92,102 used for our evaluation in Section 5.2. Feature extraction is included in time measurement.

On average, this results in a training time of 339.71 seconds ($5,66$ minutes) for an RF.

An RF is able to classify 92,102 unknown samples within 234.76 seconds. This means that on average performing classification of a single unknown sample takes 0.0025 seconds for RFs including feature extraction.

Based on the measurements presented above FANCI is able to perform classification for 400 packets per second on a general purpose computer using a single thread. As in the network of RWTH Aachen University as presented in Section 5.1 on average there are 164 NXD responses per second with a maximum peak of 900 NXD responses per second, we can state that FANCI is real-world applicable and is even able to perform live detection in large networks without sampling.

5.5 Successfully Resolved Domain Names

If a device is detected by FANCI to be infected with a bot it will ultimately successfully query for the IP address of its C2 server. If such a successful query can be detected (e.g., by using FANCI on the successful queries of infected devices after their identification), this reveals the IP address of a C2 server for the botnet in question.

We therefore present a preliminary evaluation of how well FANCI is able to separate mAGDs from success-

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.94962	0.97387	0.92537	0.02613	0.07463
σ	0.00071	0.00068	0.00108	0.00068	0.00108
x_{min}	0.94809	0.97195	0.92328	0.02508	0.07251
$\tilde{x}$	0.94973	0.97382	0.92530	0.02618	0.07470
x_{max}	0.95060	0.97492	0.92749	0.02805	0.07672

Table 12: Classification accuracy for 5-fold CV on successfully resolved domains and mAGDs of arbitrary DGAs using RFs.

fully resolving queries. In particular, we performed test measurements using random forests and a setup similar to the *mixed DGA* case presented in Section 5.2.2. Instead of bNXDs we composed the data sets of successful resolved domains from the Siemens network and known mAGDs of arbitrary DGAs. As in Section 5.2.2 we performed 5 repeated 5-fold CVs on 20 sets. Without further optimizations or new features adapted for successfully resolved domains, we achieved a mean ACC of 0.94962 with a small standard deviation of 0.00071, a minimum of 0.94809 and a maximum of 0.95060. Table 12 presents detailed results for this proof of concept experiment using RFs. Results for SVMs can be found in Appendix A.

Considering the fact that we only require to process successfully resolved domains of single devices or small groups of devices, the previously presented approach is highly promising for performing identification of C2 servers.

6 Related Work

In the past, monitoring DNS traffic (successfully resolving and/or non-resolving) has been used as primary or additional source of information in detecting malicious activity in a network (e.g., [2, 16, 18, 9, 4]). Some of these approaches have concentrated on identifying C2 servers (e.g., [18, 16]), others have focused on detecting mAGDs (e.g., [2]), identifying infected devices (e.g., [9]), or detecting malicious URLs in general (e.g., [4]).

The most striking difference between these prior approaches and FANCI is that they all require more or less extensive tracking of DNS traffic, that is, they require a correlation of information extracted from groups of DNS queries and/or responses (e.g., for features extraction). In contrast, the features that FANCI's classification module uses when predicting a particular NXD are extracted from this NXD alone, such that FANCI does not require any tracking. In addition, many of the prior approaches are based on clustering, which indulges manual labelling of the identified clusters. As opposed to this, FANCI (like [4]) makes use of an ML-classifier.

Detecting mAGDs in successfully resolving DNS traffic allows for identifying C2 servers (see Section 5.5 for an initial evaluation of FANCI in this context). However, monitoring only NXD responses has the advantage that infections with bots can be detected with less delay and while processing significantly less traffic as the vast majority of DGAs issue many more NXDs than registered names.

While the prior works show promising detection capabilities on specific data sets, little information on their generalizability and the efficiency of their detection process in terms of time and memory requirements is reported. FANCI is highly efficient with respect to both prediction (0.0025s/sample) and training (5.66min on 92102 samples) and shows a high accuracy with low FPR in very large scale realistic scenarios even when trained on a different network.

A fair comparison between FANCI and the prior approaches with respect to detection accuracy and efficiency is hard to achieve as they aim at slightly different targets and use different data sets even if they do aim at the same target. These data sets and the implementations of the systems are not publicly available. In the following, we nevertheless discuss the approaches most closely related to FANCI in more detail.

Exposure. Bilge et al. [4] introduce a system called Exposure that aims at detecting malicious domain names in DNS traffic in general, that is, they do not focus on mAGDs but also aim at detecting domain names used in the context of phishing or in the context of hosting malicious code. In contrast to FANCI, Exposure monitors full DNS traffic and not only NXD responses. Additionally, Exposure always requires access to more sensitive information than FANCI (e.g., access patterns). Like FANCI, Exposure is based on ML-classification and uses a small set of carefully selected features. However, the features are not only extracted from single domain names but also include features extracted from correlating several DNS queries or responses. The accuracy of Exposure lies in a similar range as FANCI's ACC (but targeting detecting malicious domain names in general) and is evaluated on real-world data as well. Due to requiring sensitive and contextual information, Exposure is not as versatile as FANCI especially when it comes to software-as-a-service deployments.

Winning with DNS Failures. Yadav and Reddy [18] were the first to consider the detection of botnets leveraging both DNS responses of successfully resolving domain names and NXD responses. They introduce a system primarily targeting at the identification of IP addresses of C2 servers of DGA-based botnets. The system is based on narrowing down a set of potentially malicious

IP addresses by filtering. This filtering requires access to the overall successfully resolving DNS traffic (in order to count the number of domains that resolve to a given IP address), NXD responses in the vicinity of successful queries, as well as the entropy of failed and successful DNS queries. The output of the filtering is a set of potential C2 server IP addresses.

Pleidas. Antonakakis et al. [2] present a DGA detection and discovery system called *Pleidas*. The system is able to discover new DGAs by means of clustering and to detect known DGAs by means of a supervised learning using a multi-class variant of alternating decision trees. Applying their system in a large ISP environment over a period of 15 months, they discovered twelve new DGAs, where six of them are completely new and six are variants of previously known ones.

Pleidas uses a set of statistical and structural features, where all features are extracted from groups of NXD responses originating from a single host.[9] The statistical features include entropy measures and n-grams over the group of domain names. The structural features comprise domain lengths, uniqueness and frequency distributions of TLDs, and the number of subdomain levels present.

Pleidas' classification accuracy is evaluated on labeled data. The top 10,000 domains of Alexa serve as benign class. The malicious data set consists of 60,000 NXD responses generated by four DGAs, namely *Bobax*, *Conficker*, *Sinowal*, and *Murofet*. For a group size of 5 NXD responses of each host the TPR is in the range of 95 and 99 percent and the FPR is between 0.1 and 1.4 percent. With 10 NXD responses per group, the accuracy slightly increases. In this case, the TPR is in a range of 99 and 100 percent, where the FPR ranges between 0 and 0.2 percent.

As Pleidas requires tracking of DNS responses for feature extraction, we expect that it is much less efficient than FANCI. The reported detection quality is similar to FANCI but FANCI is evaluated on a more extensive data set that uses far more DGAs and real world-benign traffic instead of the top 10,000 domains of Alexa. The generalizability of Pleidas is not evaluated.

Phoenix. Schiavoni et al. [16] present a DGA-based botnet tracking and intelligence system called Phoenix. In contrast to the previously presented Pleidas, Phoenix focuses on intelligence operations instead of DGA detection. This especially includes the tracking of C2 infrastructures of botnets regarding their IP address ranges. However, Phoenix is also capable of labeling DNS traffic as either DGA-related or benign.

They evaluated the classification performance of Phoenix on 1,153,516 domains overall including mAGDs of three different DGAs and bNXDs obtained from a passive DNS. The evaluation yielded TPRs in the range of 81.4 and 94.8 percent and is is thus significantly lower than FANCI in with respect to mAGDs detection. As the features used are less light-weight and require tracking we expect Phoenix to be less efficient than FANCI with respect to speed.

NetFlow. Grill et al. [9] present a different approach for DGA-based malware detection, with the particular goal of being applicable in large scale networks in a privacy-preserving manner. Their system is based on NetFlow data exclusively, that is, on an aggregation of metadata of network packets exchanged between a combination of a source IP and port and a destination IP address and port. The exported metadata depends on the particular implementation of NetFlow, but typically includes: IP addresses, time stamps, port numbers, byte counters, and packet counters. Grill et al. use the standardized IPFIX NetFlow format [12]. They perform an anomaly detection based on the assumption that normal behaviour of a host is to request an IP address via DNS for a certain domain name, followed by one or multiple connections to this newly resolved IP address. They assume that a DGA malware infected device is characterized by regularly issuing DNS requests without subsequent connections to new IP addresses.

For their evaluation they performed three experiments considering different types of hosts, network sizes, and times of the day. They consider six different DGAs. The ACC value is in the range of 88.77 and 99.89 percent depending on the setup in question and thus lower than FANCI's accuracy. As NetFlow is based on extensive tracking, it can be expected to be less efficient than FANCI.

DGArchive. Plohmann et al. [14] presented an extensive study of current DGAs. Their paper is based on the collection and reverse engineering of DGA-based malware and provides detailed technical insights in the functionality of modern DGAs divisible in three main contributions: a taxonomy of DGAs, a database of DGAs and corresponding mAGDs called DGArchive, and an analysis of the landscape of registered mAGDs. While Plohmann et al. do not implement an automated detection, the DGArchive provides the means to blacklist known mAGDs. Our work builds on DGArchive in two ways: we use it to clean our benign traffic before training and we use it as source for malicious mAGDs.

[9] As opposed to this, FANCI uses features extracted from individual NXDs only.

7 Conclusion

In this work, we presented FANCI, a versatile system for the detection of malicious DGA-related domain names among arbitrary NXD DNS traffic based on supervised learning classifiers. FANCI's versatility is a result of its lightweight and language independent feature design relying exclusively on domain names for classification. In our extensive evaluation, we verified FANCI's highly accurate and highly efficient detection capabilities of mAGDs in different experiments, including its generalizability. In an one-month real-world application in a large university network, we were able to discover ten new DGA-related groups of mAGDs, where at least four of them originate from brand new DGAs.

With its empirically proven detection capabilities and a successful real-world test, FANCI can make a decisive contribution to combating DGA-based botnets. FANCI is able to provide valuable information to existing security solutions and is able to contribute to a higher level device and network security in a variety of environments.

Acknowledgements

We would like to thank Daniel Plohmann for granting us access to DGArchive. Many thanks to Jens Hektor and Thomas Penteker for providing us NXD data from RWTH Aachen University and Siemens respecitvely. Thanks to the ITCenter of RWTH Aachen University for granting us extensive access to the university's compute cluster.

References

[1] ANTONAKAKIS, M., PERDISCI, R., DAGON, D., LEE, W., AND FEAMSTER, N. Building a Dynamic Reputation System for DNS. In *19th USENIX security symposium* (2010), USENIX Association, pp. 273–290.

[2] ANTONAKAKIS, M., PERDISCI, R., NADJI, Y., VASILOGLOU II, N., ABU-NIMEH, S., LEE, W., AND DAGON, D. From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware. In *21th USENIX security symposium* (2012).

[3] AUTHORITY, I. A. N. IANA list of top-level domains, July 2017.

[4] BILGE, L., SEN, S., BALZAROTTI, D., KIRDA, E., AND KRUEGEL, C. Exposure: A Passive DNS Analysis Service to Detect and Report Malicious Domains. *ACM Trans. Inf. Syst. Secur.* (Apr. 2014), 14:1–14:28.

[5] BREIMAN, L. Bagging predictors. *Machine Learning* (Aug. 1996), 123–140.

[6] BREIMAN, L. Random Forests. *Machine Learning* (Oct. 2001), 5–32.

[7] CORTES, C., AND VAPNIK, V. Support-vector networks. *Machine Learning* (Sept. 1995), 273–297.

[8] FOUNDATION, M. Public Suffix List, Apr. 2017.

[9] GRILL, M., NIKOLAEV, I., VALEROS, V., AND REHAK, M. Detecting DGA malware using NetFlow. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)* (May 2015), pp. 1304–1309.

[10] HO, T. K. Random Decision Forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition* (Washington, USA, 1995), ICDAR, IEEE Computer Society.

[11] ICANN. ICANN Research - TLD DNSSEC Report, Feb. 2017.

[12] J. QUITTEK, T. ZSEBY, B. C. S. Z. Requirements for IP Flow Information Export (IPFIX). RFC 3917, IETF, October 2004.

[13] NUMBERS, I. C. F. A. N. A. Registry Listing - ICANN, Apr. 2017.

[14] PLOHMANN, D., YAKDAN, K., KLATT, M., BADER, J., AND GERHARDS-PADILLA, E. A comprehensive measurement study of domain generating malware. In *25th USENIX Security Symposium* (Austin, TX, 2016), USENIX Association, pp. 263–278.

[15] RWTH AACHEN UNIVERSITY, I. C. Statusmeldungen zentraler Systeme - RWTH AACHEN UNIVERSITY IT Center - Deutsch, Aug. 2017.

[16] SCHIAVONI, S., MAGGI, F., CAVALLARO, L., AND ZANERO, S. Phoenix: DGA-Based Botnet Tracking and Intelligence. In *Detection of Intrusions and Malware, and Vulnerability Assessment* (July 2014), Springer, Cham, pp. 192–211.

[17] SOPHOS. Sophos Live Protection: Overview, Aug. 2017.

[18] YADAV, S., AND REDDY, A. L. N. Winning with DNS Failures: Strategies for Faster Botnet Detection. In *Security and Privacy in Communication Networks* (Sept. 2011), Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer, Berlin, Heidelberg, pp. 446–459.

[19] ZDRNJA, B. Google Chrome and (weird) DNS requests, Aug. 2017.

A Results for SVMs

In this section, we present results for SVMs for the experiments presented in Section 5.2.2, Section 5.2.3, and Section 5.5.

	ACC	**TPR**	**TNR**	**FNR**	**FPR**
$\bar{x}$	0.99930	0.99983	0.99878	0.00017	0.00122
σ	0.00190	0.00103	0.00331	0.00103	0.00331
x_{min}	0.98133	0.99188	0.96400	0.00000	0.00000
$\tilde{x}$	0.99971	1.00000	0.99942	0.00000	0.00058
x_{max}	1.00000	1.00000	1.00000	0.00812	0.03600

Table 13: Results for classifying bNXDs and mAGDs of single DGAs with SVMs. In total, 295 sets of 59 DGAs were considered each evaluated by 5 repetitions of a 5-fold CV.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.98315	0.96713	0.99916	0.03139	0.00084
σ	0.06166	0.12291	0.00085	0.11956	0.00085
x_{min}	0.49850	0.00000	0.99564	0.00000	0.00000
$\tilde{x}$	0.99965	1.00000	0.99935	0.00000	0.00065
x_{max}	1.00000	1.00000	1.00000	1.00000	0.00436

Table 14: Results for LOGO CV for mAGDs of single DGAs grouped by seed using SVMs. In total, 150 sets of 30 DGAs were considered.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.99464	0.99148	0.99779	0.00852	0.00221
σ	0.00017	0.00056	0.00037	0.00056	0.00037
x_{min}	0.99430	0.99037	0.99721	0.00755	0.00146
$\tilde{x}$	0.99468	0.99156	0.99784	0.00844	0.00216
x_{max}	0.99492	0.99245	0.99854	0.00963	0.00279

Table 15: Results for detecting mAGDs with SVMs of arbitrary mixed DGAs using 5 repetitions of 5-fold CV for each set. In total, 20 sets were considered.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.97972	0.96195	0.99746	0.02635	0.00254
σ	0.00041	0.00056	0.00040	0.00061	0.00040
x_{min}	0.97894	0.96088	0.99672	0.02517	0.00161
$\tilde{x}$	0.97967	0.96207	0.99747	0.02622	0.00253
x_{max}	0.98073	0.96304	0.99839	0.02751	0.00328

Table 16: Results for LOGO CV for sets of mAGDs of mixed DGAs grouped by DGA using SVMs. In total, 20 sets were considered.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.99394	0.99331	0.99456	0.00669	0.00544
σ	0.00031	0.00070	0.00047	0.00070	0.00047
x_{min}	0.99327	0.99135	0.99371	0.00575	0.00467
$\tilde{x}$	0.99402	0.99341	0.99451	0.00659	0.00549
x_{max}	0.99436	0.99425	0.99533	0.00865	0.00629

Table 17: Results for classifying mAGDs of arbitrary mixed DGAs and bNXD from Siemens applying 5 repetitions of 5-fold CV for 20 sets each of size 100,000 using SVMs.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.99180	0.99252	0.99108	0.00748	0.00892
σ	0.00026	0.00014	0.00047	0.00014	0.00047
x_{min}	0.99133	0.99211	0.99016	0.00728	0.00793
$\tilde{x}$	0.99185	0.99254	0.99112	0.00746	0.00888
x_{max}	0.99240	0.99272	0.99207	0.00789	0.00984

Table 18: Classification accuracy for training on RWTH Aachen data and prediction on Siemens data using SVMs.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.99448	0.99412	0.99485	0.00588	0.00515
σ	0.00017	0.00017	0.00033	0.00017	0.00033
x_{min}	0.99419	0.99387	0.99432	0.00558	0.00441
$\tilde{x}$	0.99447	0.99415	0.99483	0.00585	0.00517
x_{max}	0.99479	0.99442	0.99559	0.00613	0.00568

Table 19: Classification accuracy for training on Siemens data and prediction on RWTH Aachen data using SVMs.

	ACC	TPR	TNR	FNR	FPR
$\bar{x}$	0.93683	0.98900	0.88465	0.01100	0.11535
σ	0.00059	0.00049	0.00103	0.00049	0.00103
x_{min}	0.93565	0.98807	0.88269	0.00990	0.11371
$\tilde{x}$	0.93689	0.98913	0.88470	0.01087	0.11530
x_{max}	0.93778	0.99010	0.88629	0.01193	0.11731

Table 20: Classification accuracy for 5-fold CV on successfully resolved domains and mAGDs of arbitrary DGAs using SVMs.

B Grid Search Results

In this section, we present results for our grid search. To reduce the number of grid searches that have to be performed for the *single-DGA detection*, we only did one grid search per DGA generation scheme as introduced in the taxonomy by Plohmann et al. [14]. We performed all grid searches on sets of size 20,000. To avoid overfitting we performed grid searches on 6 independent sets for the *multi-DGA detection* case. The final parameter selection for *multi-DGA detection* is based on mathematical constraints of the respective ML algorithm and on domain knowledge on the classification problem. The ML algorithm parameters are named according to standard references for SVMs [7] and RFs [6].

For RFs we performed one grid search per data set as follows. Parameter T is an integer drawn uniformly at random from $[10, 1000]$, where we considered 64 values for T in total. As our feature vector is of length 44, F is an integer selected from $[2, 44]$, where each possible value is assigned to F. The impurity criterion $i(N)$ is either Gini impurity or entropy impurity. This results in $64 \cdot 43 \cdot 2 = 5504$ 5-fold CVs in total per data set.

For SVMs we performed one grid search per data set as follows. After some initial tests we fixed the parameter range for C and γ to $[2^{-16}, 2^{3}]$ and considered 80 values drawn logarithmically at random for both parameters. This results in 80 5-fold CVs for the linear kernel and in $80^2 = 6400$ 5-fold CVs for the RBF kernel per data set.

The following tables present the resulting best parameter choices according to the ACC.

Set #	i(N)	F	T	ACC
1	entropy	25	17	0.9981
2	Gini	10	33	0.9993
3	entropy	22	72	0.9983
4	Gini	7	161	0.9987
5	Gini	13	227	0.9984
6	Gini	31	785	0.9983
Final	**Gini**	**18**	**785**	—

Table 21: Best parameter choices for independent data sets of mixed DGAs for RFs. For the final selection $i(N)$ is selected by majority vote. F is the arithmetic mean. For T the maximum is chosen.

Gen. Scheme	DGA	i(N)	F	T	ACC
Arithmetic	Corebot	Gini	8	681	0.9999
Hash	Dyre	Gini	2	388	1.0
Wordlist	Matsnu	Gini	5	57	0.9999
Permutation	VolatileCedar	Gini	2	513	1.0

Table 22: Best parameter choices depending on the generation scheme of the DGA for RFs. The above parameters are used among all experiments where single DGAs are considered and are applied depending on the DGA's generation scheme.

Set #	Kernel	C	γ	ACC
1	RBF	2.9423	0.0198	0.9992
2	linear	0.1729	—	0.9982
3	RBF	1.7844	0.0102	0.9985
4	RBF	2.9423	0.0234	0.9982
5	RBF	4.8517	0.0073	0.9982
6	RBF	5.7317	0.0751	0.9979
Final	**RBF**	**0.9160**	**0.0198**	—

Table 23: Best parameter choices for independent data sets of mixed DGAs for SVMs. For the final selection the kernel is selected by majority vote. C is selected as median. γ is chosen as the arithmetic mean. Both only among the RBF results.

Gen. Scheme	DGA	Kernel	C	γ	ACC
Arithmetic	Corebot	linear	3.4669	—	0.9999
Hash	Dyre	linear	0.0052	—	1.0
Wordlist	Matsnu	linear	0.2289	—	0.9999
Permutation	VolatileCedar	RBF	0.0234	0.0327	1.0

Table 24: Best parameter choices depending on the type of DGA for SVMs. The above parameters are used among all experiments where single DGAs are considered and are applied depending on the DGA's generation scheme.