Neural Reputation Models learned from Passive DNS Data

Pierre Lison
Norwegian Computing Center
Oslo, Norway
Email: plison@nr.no

Vasileios Mavroeidis
Department of Informatics
University of Oslo
Oslo, Norway
Email: vasileim@ifi.uio.no

***Abstract*—Blacklists and whitelists are often employed to filter outgoing and incoming traffic on computer networks. One central function of these lists is to mitigate the security risks posed by malware threats by associating a *reputation* (for instance benign or malicious) to end-point hosts. The creation and maintenance of these lists is a complex and time-consuming process for security experts. As a consequence, blacklists and whitelists are prone to various errors, inconsistencies and omissions, as only a tiny fraction of end-point hosts are effectively covered by the reputation lists. In this paper, we present a machine learning model that is able to automatically detect whether domain names and IP addresses are benign, malicious or sinkholes. The model relies on a deep neural architecture and is trained on a large passive DNS database. Evaluation results demonstrate the effectiveness of the approach, as the model is able to detect malicious DNS records with a F_1 score of 0.96. In other words, the model is able to detect 95 % of the malicious hosts with a false positive rate of 1:1000.**

***Keywords*-DNS reputation, malware, neural networks, passive DNS, machine learning, graph inference, cybersecurity**

I. Introduction

Given the growing risks posed by the current cyber-threat landscape, the deployment of effective techniques for detecting malicious activities becomes increasingly important for both public and private organisations. In their 2017 Data Breach Investigation Report[1], Verizon observes that the majority of investigated breaches included some form of malware, with backdoor and command & control (C2) mechanisms being one of the most prominent hacking varieties. Traditional defence strategies such as reputation lists of domain names and IP addresses are often employed to block communication channels serving malicious purposes. However, these defence strategies can be circumvented relatively easily by threat agents through techniques such as fast-flux networks. Fast flux networks operate by changing DNS records at high speed in order to evade static blacklists and resist takedown attempts.

In this work, we present an alternative, data-driven approach to the detection of malicious end-point hosts. Based on a large passive DNS dataset, we demonstrate how a deep learning architecture can be used to automatically predict the reputation of DNS records with high accuracy. The approach presents multiple benefits compared to traditional reputation lists. The most important advantage is the model ability to provide predictions in real-time, without human intervention. This enables faster and more effective responses to cyber-attacks. The model is also less vulnerable to human errors and omissions than traditional reputation lists (which must be regularly updated by security experts).

The rest of this paper is as follows. The next section outlines the key principles behind dynamic reputation models and the most important approaches developed in previous work. Section III details the various data sources employed to train the reputation models. Section IV describes how to capture information about neighbouring hosts through graph inference, and Section V presents the neural architecture used to predict the reputation of end-point hosts. The evaluation results are reported in Section VI, along with a short discussion. Finally, Section VII concludes this work.

II. Background

Several prior studies have investigated the use of passive DNS monitoring for the identification of malicious domains. Antonakakis et al. (2010) described Notos, a dynamic reputation system based on the observation that malicious uses of DNS have unique characteristics, distinguishable from legitimate professionally provisioned DNS services. Notos employs a broad range of features, which can be network-based (number of total IPs historically associated with a domain, diversity of geographical locations, number of distinct autonomous systems in which they reside, etc.), zone-based (average length of domain names in related domains, number of distinct top-level domains, character frequencies, etc.), and evidence-based (number of malware samples that contacted the domain or that are connected to an IP pointed to by the domain).

Bilge et al. (2014) describe Exposure, a system similar to Notos but requiring less training time and data. Exposure is able to overcome some of the limitations of Notos, as it is able to identify malicious domains and addresses that were never seen in malicious activities before. Their system uses 15 features extracted from DNS traffic, allowing them to characterise different properties of domain names and how these are queried. Specifically, the features of Exposure are either time-based (short life, daily similarity,

[1] http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/

repeating patterns, access ratio), DNS answer-based (number of distinct IP addresses, number of distinct countries, number of domains sharing the IP address, reverse DNS query results), TTL value-based (average TTL, standard deviation of TTL, number of distinct TTL values, number of TTL change), and domain name-based (percentage of numerical characters and normalised length of the longest meaningful substring). In a real world evaluation of 2 weeks, their system identified 3000 previously unknown malicious domains without generating any false positive.

In contrast to Notos and Exposure, which both rely on monitoring traffic from local recursive DNS servers, the Kopis system Antonakakis et al. (2011) uses passive DNS data aggregated at the upper levels of the DNS hierarchy and can detect malware domains even when no IP reputation information is present, by analysing global DNS query resolution patterns. Kopis divides the monitored data streams into epochs and summarises the DNS traffic for a given domain name at the end of each epoch by computing a number of statistical features, such as the diversity of the IP addresses associated with the recursive DNS servers that queried a specific domain, the relative volume of queries from querying recursive DNS servers, and historic information related to the IP space pointed to by the domain.

Khalil et al. (2016) argue that many local features used in detecting malicious domains, such as domain name and temporal patterns tend to be relatively brittle and allow attackers to take advantages of these features to evade detection. To address this issue, they developed graphs reflecting the global associations among domains and IPs, and they proposed a path-based mechanism to derive a malicious score at each domain based on their topological connection to known malicious domains.

Peng et al. (2017) proposed a malicious domain detection method focusing on domains that are not resolved to IP addresses directly, but only appear in DNS CNAME records, based on the idea that the domains connected by CNAME resource records share intrinsic relations and are likely to be similar to one another. However, the authors observe that there exists scenarios where a domain is CNAMEd by many other malicious domains but itself is still benign. Their approach relies on a graph-based inference technique that summarises the global association for each domain and belief propagation to compute the malicious marginal probability for each node based on its global associations with other known malicious and benign domains. Their experimental results show that their proposal can effectively uncover the malicious domains omitted by previous works based on passive DNS relying only on A records.

Finally, Watkins et al. (2017) proposed a semi-supervised machine learning approach to filter out non-malicious domains in passive DNS data. Their approach relies on clustering algorithms to cluster around the DNS-name based, TTL-value based, and DNS query answer-based behaviour of known malicious domains that become the reduced dataset of suspicious results.

Passive DNS data can also contribute to the identification of malicious domains generated by domain generation algorithms (DGAs). Antonakakis et al. (2012) describe a technique to detect DGAs based on the idea that bots from the same botnet (same DGA) will generate similar non-existent domain traffic ("NXDomain"). Using a combination of clustering and classification algorithms combined with real-word DNS traffic, the authors were able to discover twelve DGAs (half were variants of known DGAs and the other half new DGAs that have never been reported before). Zhou et al. (2013) presented a DGA-detection approach based on NXDOMAIN (non-existent domain) traffic. Their approach is based on the idea that every domain name in the domain group generated by one botnet using DGAs is often used for a short period of time (active time) and has similar life and query style. In addition, they group domain names by creating clusters with the same second- and third-level domains and IP addresses, and calculate domain access similarity (life time span and visit time patterns) for each group to output a suspicious DGA-domain name list.

This paper extends the approach developed by Antonakakis et al. (2010, 2011, 2012) in several directions. Most importantly, we adopt a neural approach to the problem of predicting the reputation of a given domain name or IP address. The use of deep neural architectures provides several important benefits compared to traditional, "shallow" machine learning techniques, such as the ability to capture complex patterns in the domain names (using recurrent neural networks) and handle sparse features (such as the geolocation of IP addresses) through embeddings. This reliance on neural models substantially reduce the need for handcrafted feature engineering while providing state-of-the-art classification results. Another contribution of this paper, inspired by the work of Peng et al. (2017); Watkins et al. (2017), is the use of graph-based features to exploit relations between neighbouring domain names and IP addresses.

III. Dataset

A. Passive DNS

The passive DNS data used in this work was kindly provided by Mnemonic[2]. The raw dataset consists of 567 million aggregated DNS queries collected over a period of four years. Each entry is defined by the following variables:

- A record type (the most common DNS records in this dataset are A, CNAME, AAAA or PTR records)
- A recorded query
- An answer to the above query
- A Time-to-Live (TTL) value for the query-answer pair
- A timestamp for the first occurrence of the pair
- A timestamp for the last occurrence of the pair

[2] https://passivedns.mnemonic.no

- The total number of occurrences of the pair during the period the data was collected.

The majority of the DNS entries are A records – that is, records mapping domain names to their corresponding IP addresses. There is a total of 476 million A records in the dataset (84 % of the total number of entries). There is also about 63 million CNAME records (11 % of the total), whose function is to provide aliases between domain names. The remaining records are made of AAAA records (4 % of the total) and PTR records (1 % of the total). For the purpose of this paper, we shall focus on the A, AAAA and CNAME records and discard the other entries, as they are not directly relevant to the reputation of domain names and IP addresses.

Based on A and AAAA records, we extract the following information for each distinct ⟨domain, IP address⟩ record:

- The domain name, divided in top-level and second-level domains and optionally a subdomain ;
- The IP address, either in IPv4 or IPv6 ;
- The number of TTL changes observed for this pair ;
- The minimum TTL value observed for this pair ;
- The total number of queries for the pair.

This extraction process results in a total of 171 million distinct domain names and 17 million IP addresses. Each domain name is resolved to an average of 2.21 IP addresses, with a large standard deviation $\sigma = 18.3$. This large standard deviation is due to the fact that some domains are resolved to several thousand IP addresses. Similarly, each IP address is hosting an average of 22.6 domain names, again with a large standard deviation $\sigma = 1864$ (some IP addresses are used by as many as two million domain names).

In complement to this dataset of ⟨domain, IP address⟩ records, a table of domain aliases was also extracted based on the CNAME records present in the passive DNS data. This table was used to compute the number of aliases associated with each domain name and used as a feature for the neural network model presented in Section V.

B. Reputation labels

To apply supervised learning to the problem of predicting the reputation of domain names and IP addresses, we must associate a portion of the dataset with reputation labels. These reputation labels can fortunately be automatically extracted from existing blacklists and whitelists. The domain names and IP addresses were labelled with 4 reputation values: **unknown**, **benign**, **malicious** or **sinkhole**.

Table I provides a summary of the most important statistics regarding the dataset used for this work.

1) Domain whitelists:

We downloaded eight snapshots of the Alexa Top 1 million domains (spread from 2010 to 2017), along with similar whitelists such as the top 1 million from Statvoo and Cisco. It should be noted that these rankings only enumerate popular domains and offers no guarantee that the domains are malware-free. However, in practice, malicious domains have a low probability of appearing on these ranked lists, as malware domains are by nature transient and are very unlikely to stay on a top list of popular domains for an extended period of time. These whitelists are subsequently augmented by two whitelists, one from maltrail[3] and one in-house whitelist from Mnemonic.

This extraction process resulted in a list of 4 million domains marked as benign. We applied this list on the passive DNS dataset and were able to label a total of 39 million domains (making up 22 % of the total number of domain names) as benign. This number is higher than the original list of 4M benign domains due to the fact that the passive DNS data contains complete domain names (including subdomains) while whitelists cover only the top and second-level parts of the domain name.

2) Domain blacklists:

The identification of malicious domains in the dataset followed a similar procedure. Three reputable sources were employed:

1) The blacklist from maltrail (which is itself a compilation of blacklists obtained from various sources such as alienvault), totalling 1.3 million domains ;
2) An in-house domain blacklist provided by Mnemonic with an additional 100 thousand domains ;
3) 2.9 million malware domains produced by domain-generating algorithms (DGAs) from DGArchive[4].

After merging these blacklists into one unified list, we obtained a collection of 4.2 million malicious domains. Upon applying this list on the passive DNS dataset, we find a total of 2.8 million domains labelled as malicious.

In addition, blacklists also provide succinct information about the type of malicious activity associated with the domain (spam, fishing, malware, etc.) along with a confidence level (for instance, many blacklists include the word "suspicious" when the the domain is not confirmed malicious). Based on these descriptions, we enriched the reputation labels with a *malicious category* (comprising 454 distinct classes) and a *confidence level* (low or high).

3) IP whitelists and blacklists:

The white- and black-lists of IP addresses are extracted from existing lists provided by maltrail and from an in-house list from Mnemonic. This resulted in a list of 69 thousand benign IP addresses (or sub-networks) and 210 thousand malicious IP addresses (or sub-networks). These two lists were then applied on the IP addresses from the passive DNS data, leading to 54 thousand IP addresses labelled as benign (0.33 %) and 1.2 million IP addresses labelled as malicious (7.5 %). Using an approach similar to the

[3]https://github.com/stamparm/maltrail

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

Description	Number of occurrences	
Number of distinct domain names	171 106 318	
Number of *benign* domains	38 811 436	(21.65 %)
Number of *malicious* domains	2 903 996	(1.62 %)
... of which marked as high confidence	1 545 902	(53 % of above)
Number of *sinkhole* domains	14858	(0.008 %)
Number of distinct *types* of malicious domains	454	
Number of distinct IP addresses	16 768 026	
Number of *benign* IP addresses	56 636	(0.33 %)
Number of *malicious* IP addresses	1 259 214	(7.5 %)
... of which marked as high confidence	16 228	(1.3 % of above)
Number of *sinkhole* IP addresses	291	(0.002 %)
Number of distinct *types* of malicious addresses	72	
Number of IP addresses with known geolocation and ISP	16 715 799	(99.7 %)
Number of distinct geolocations covered by the IP addresses	127 399	
Number of distinct (ISPs) covered the IP addresses	490 945	
Number of distinct $\langle$ domain, IP address$\rangle$ records	378 171 968	
Number of *benign* records	122 177 956	(32.3 %)
Number of *malicious* records	9 275 476	(0.26 %)
Number of *sinkhole* records	201 461	(0.05 %)
Average out-degree for domain names	2.21	(std = 18.3)
Average out-degree for IP addresses	22.6	(std = 1864)
Number of $\langle$domain, domain $\rangle$ aliases	40 321 236	

Table I: Summary statistics regarding the passive DNS data.

one used for the domain names, the malicious IP addresses were marked as either high confidence (confirmed malware) or low confidence (i.e. suspicious). Only a small fraction (1.3 %) of the IP addresses marked as malicious has high confidence. The remaining parts consist of IP addresses that are not confirmed malicious but are part of networks known for hosting suspicious activities and botnets, as identified by their Autonomous System Number (ASN).

4) Sinkholes:

In addition to benign and malicious reputations, we also labelled some of the domains and IP addresses with a specific label for *sinkholes*. DNS sinkholes intercept DNS requests to known malicious domains and redirects them to benign IP addresses, where it can be further analyzed by experts and/or law enforcement officials (Bruneau and Wanner, 2010). Introducing sinkholes as an explicit reputation category (besides benign and malicious) provides us with a useful source of information for the detection of malicious domains. Indeed, the fact that a domain name was at some point redirected to a sinkhole IP address is a strong indication that the domain was associated with a malicious activity in the days or weeks preceding the redirection.

We compiled two small lists of sinkhole domains (54 instances) and sinkhole sub-networks (987 instances) and applied them on the passive DNS dataset, leading to 1.4 thousand domains and 272 IP addresses labelled as sinkhole.

5) Reputation of $\langle$domain, IP address$\rangle$ records:

Although the reputation labels of domain names and IP addresses extracted from white- and blacklists are undeniably useful, they are also prone to errors and inaccuracies. To increase the quality of the training data used for estimating the reputation models, we employed the following procedure to determine the reputation of each record. Let R_{dom} be the reputation of the domain name and R_{ip} the reputation of the associated IP address. These reputations can have four possible values: unknown, benign, malicious or sinkhole. The reputation of the $\langle \text{dom}, ip \rangle$ record is then defined as:

1) If R_{dom} and R_{ip} are known and $R_{dom} \neq R_{ip}$ (in other words, there is a conflict between the reputations of the domain name and its associated IP address), the reputation of the record is marked as unknown.
2) If $R_{dom} = R_{ip}$, the reputation of the record is labelled with the same reputation.
3) If either R_{dom} or R_{ip} is unknown and the other reputation is marked as high confidence, the reputation of the record is labelled with this reputation.
4) In all remaining cases (i.e. when the two reputations are either unknown or marked as being of low confidence), the reputation is marked as unknown.

C. Geolocation

An important factor influencing the reputation of IP addresses is their geographical location and Internet Service

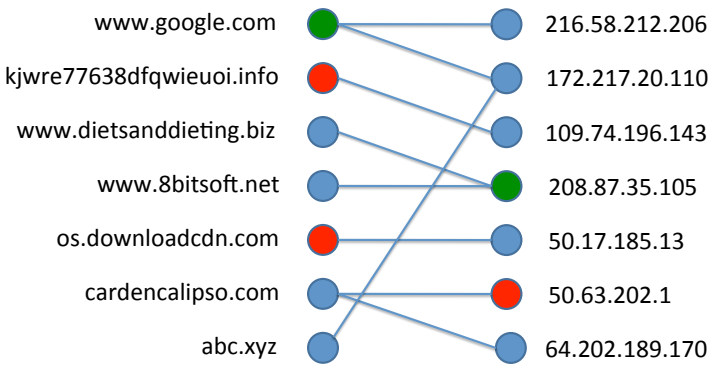

Figure 1: Illustration of a bipartite graph extracted from passive DNS data. Green circles indicate a benign reputation, red circles a malicious reputation, and blue circles an unknown reputation.

Provider (ISP). In order to exploit this intelligence source, we acquired a large dataset of IP geolocations[5] and used it to annotate the IP addresses with both a *geoname* identifier and the name of their ISP. 99.7 % of the IP addresses could be annotated using this dataset.

The GeoNames database[6] covers over 11 million placenames. The resolution of these placenames is quite fine-grained, sometimes as precise as city blocks. Each geoname identifier is provided with various geographical information, such its country, state or province, city, longitude and latitude. More than 127 thousand distinct placenames and 490 thousand distinct ISPs were found to be associated with the IP addresses present in the passive DNS data.

IV. Graph inference

As detailed in Deri et al. (2013), DNS traffic can be represented as a large graph, more precisely as a *bipartite graph*. A bipartite graph is a graph whose vertices can be divided into two disjoint and independent sets such that every edge connects one vertex from the first set with one vertex from the second set. In our case, the first set corresponds to the domain names, and the second set corresponds to IP addresses. Figure 1 illustrates such a bipartite graph structure. We focus here on A and AAAA records, which are the most important source of information regarding the reputation of domain names and IP addresses. The inclusion of CNAME records of domain aliases in the graph is of course possible but would increase the complexity of the inferential process by breaking the bipartite structure of the graph.

The graph structure can provide important clues about the reputation of a given record. For instance, in Figure 1, the reputation of the domain name abc.xyz is unknown. However, one of its IP addresses is connected to another domain name (www.google.com) which is known as benign. Similarly, the IP address 64.202.189.170 has an unknown reputation, but is connected to a domain name that was itself connected to an IP address marked as malicious. The reputation of a given node (domain name or IP address) may therefore be influenced by the reputation of its neighbours in the bipartite graph. This influence is inversely proportional to the distance between the neighbour and the node (the closer the node, the larger the influence).

Extracting the local neighbours of all nodes in the graph and aggregating their reputation is, however, far from trivial due to the sheer size and connectedness of the passive DNS dataset. A naive traversal of the graph is not scalable, as it may take weeks or months to complete. An alternative approach is to split the nodes in subgroups and run a separate traversal algorithm for each subgroup in parallel. However, this parallelisation is difficult in practice due to the large memory requirements necessary for storing the bipartite graph and the associated reputations of each node.

The approach taken is this work is to rely on graph theory and represent the bipartite graph as a large, sparse $|D| \times |I|$ matrix R, where $|D|$ is the number of distinct domain names and $|I|$ the number of distinct IP addresses in the dataset. This matrix is called an *adjacency matrix*. Each cell (d, i) in the matrix has only two possible values: 1 if there exists a record between the domain d and the IP address i in the dataset, and 0 otherwise. This matrix is very sparse, as only a tiny fraction of the potential combinations of domain names and IP addresses actually exists as records.

Based on this matrix, determining the neighbours at distance k can be expressed through matrix multiplication. The distance between nodes is expressed in terms of number of edges in the bipartite graph – for instance, www.google.com and abc.xyz have a distance of 2. To find all neighbours of domain names at distance 2, we compute the product of the adjacency matrix with its transpose, take its sign function (to reduce all positive integers to 1), and remove the diagonal (to avoid counting the node as its own neighbour):

$$N_{k=2} = \text{sign}(R \cdot R^T) - I_{|D|} \quad (1)$$

[5] https://db-ip.com/db/

[6] http://www.geonames.org

where $sign$ is the sign function and $I_{|D|}$ is the identity matrix. The $N_{k=2}$ matrix is in this case a $|D| \times |D|$ matrix where each (d_1, d_2) cell has a value 1 if d_1 and d_2 are separated by exactly 2 edges (and $d_1 \neq d_2$), and 0 otherwise. More generally, the neighbours of domain names at distance k can be calculated through a sequence of k matrix multiplications:

$$N_k = \text{sign}(\underbrace{R \cdot R^T \cdot R \cdot R^T \cdot ...}_{k \text{ matrix multiplications}}) - I_{|D|} \quad (2)$$

Based on this neighbour matrix N_k, calculating the number of malicious neighbours at a distance k of the domain names can also be computed through algebraic manipulations. Given the bipartite nature of the graph, we know that if k is an even number, all neighbours at a distance k will also be domain names, while if k is odd, all neighbours will be IP addresses. Therefore, if k is even, we create an array M of length $|D|$ with a value of 1 if the domain is malicious and 0 otherwise. If k is odd, we create a similar array M of length $|I|$ with a value of 1 if the IP address is malicious and 0 otherwise. The number of malicious neighbours M_k a distance k is then simply provided by:

$$M_k = N_k \cdot M \quad (3)$$

The array M_K will also be of size $|D|$, and each position in this array will correspond to a domain name and express the number of malicious nodes at distance k of this domain.

The same calculations can be employed to calculate the number of benign or sinkhole neighbours. To optimise these matrix operations (and limit the memory usage), we represent the matrix R using a sparse matrix format (compressed sparse row), and rely on high-performance matrix-matrix and matrix-vector multiplications (Williams et al., 2009). To further improve the performance of the matrix products, it is also possible to extract the connected components of the matrix R (that is, the set of subgraphs that are isolated from one another) and then perform separate calculations in each.

Once the reputations of neighbouring nodes is calculated, they can be used as features for the machine learning model, as described in the next section.

V. Neural model

The neural model learns to classify ⟨domain name, IP address⟩ records into three categories: **benign**, **malicious** and **sinkhole**. Several types of features can be exploited for this classification task, including both numerical features, categorical features and one sequence (the characters making up the domain name). The list of features used as inputs to the neural model is provided in Table II.

Numerical features include values such as the number of TTL changes, the lifespan of the record or the number of malicious neighbours at distance 2. As the range of possible values for some of these features can be large, the feature values are rescaled (by removing the mean and scaling to unit variance) before being fed to the neural model.

Categorical features encompass values such as the ISP associated with the IP address or the Top-Level Domain (TLD) of the domain. While these features can be very informative for the classification, their inclusion in the neural model is far from straightforward, as the number of possible classes for most categories is very large. There is for instance over 490K distinct ISPs in the database. A "one-hot encoding" of these features is therefore not directly applicable. A more scalable approach, which is used in the neural architecture developed in this paper, is to use learnable embeddings to convert each category into a dense vector (Goldberg, 2016). One advantage of embedding models is their ability to express similarities between categories – for instance, the vector for the TLD "nl" might be closer in vector space to the TLD "de" than to "cn", since the distributional properties of Dutch domains are expected to be more similar to German domains than Chinese ones. These embeddings are learned simultaneously with the rest of the neural model.

Finally, the domain name needs to be accounted for in a specific manner due to its sequential structure. The sequence of characters making up the domain name can be a very useful source of information for predicting the reputation of the domain, as many malicious domains are generated by so-called domain-generating algorithms and have distributional patterns that are different from real domain names.

The sequential structure of the domain can be captured by a recurrent layer, for instance with LSTM or GRU units (Goodfellow et al., 2016). Such a recurrent layer operates by incrementally updating a hidden state (expressed as a fixed-size vector) based on the sequence of inputs – in this case, the characters making up the domain name. Each unit in this recurrent layer takes as input one character and the previous state, and outputs an updated vector representing the state of the sequence so far. Once the last character is processed, the final output vector is used to predict whether the domain is likely to have been generated by a malware. This sub-network is trained separately from the rest of the architecture (since it can be directly estimated from a dataset of malware domains). Due to space constraints, we do not present the details of this part of the architecture in the present paper, but refer to Lison and Mavroeidis (2017) for details.

The complete neural architecture is illustrated in Figure 2. The network comprises both numerical, categorical and sequential features. Categorical features are converted into low-dimensional embeddings, and the characters making up the domain name are fed into a recurrent network that outputs a probability value expressing whether the domain is likely to be malware-generated. The numerical features, embedding vectors and probability value from the recurrent network are then combined and fed into two dense feed-forward layers (with rectified linear units as activation function). The output of the second dense layer is then applied

Numerical features:	
nb_queries	Total nb. of queries observed in the passive DNS data for the ⟨domain, IP address⟩ pair
min_ttl	Minimum TTL value for the pair
ttl_changes	Nb. of change of TTL values for the pair
lifespan	Time (in seconds) elapsed between the first and the last observation of the pair
nb_domain_queries	Total nb. of queries for the domain (aggregated over all IP addresses resolved to it)
domain_lifespan	Time (in seconds) elapsed between the first and the last observation of the domain
domain_inactivity	Time (in seconds) elapsed since the last observation of the domain
nb_ips	Nb. of distinct IP addresses that have been resolved to the domain
nb_locations	Nb. of distinct geolocations (identified by geoname identifier) where the domain was resolved
nb_isps	Nb. of distinct Internet Service Providers where the domain was resolved
nb_countries	Nb. of countries where the domain was resolved
nb_aliases	Nb. of aliases for the domain
nb_address_queries	Total nb. of queries observed for the IP address (aggregated over all domains resolving to it)
address_lifespan	Time (in seconds) elapsed between the first and the last observation of the IP address
ranking_domain	Average ranking of the domain in Alexa rankings
is_ranked	1 if the domain is ranked on Alexa, else 0
nb_domains	Nb. of domains resolved to this IP address
nb_neighbours(d)	Nb. of records at distance d of the current record
nb_benign_neighbours(d)	Nb. of records marked as benign and at distance d of the current record
nb_malicious_neighbours(d)	Nb. of records marked as malicious and at distance d of the current record
nb_sinkhole_neighbours(d)	Nb. of records marked as sinkhole and at distance d of the current record
Categorical features:	
city	City associated with the geolocation of the IP address
country	Country associated with the geolocation
stateprov	State or province associated with the geolocation
geoname_id	Geoname identifier for the IP address
ip_2bytes	First 2 bytes of the IP address
isp_id	Internet Service Provider for the IP address
suffix	Top-level domain of the domain name
Sequence:	
domain	Domain name of the record (sequence of characters)

Table II: Feature set used for the predicting the reputation of ⟨domain, IP address⟩ records.

to produce a probability distribution over the three possible reputation labels (benign, malicious or sinkhole).

VI. Evaluation

We report in this section the experimental results obtained when evaluating the classification performance of the neural network. For this evaluation, we used the full set of 378M records extracted from the passive DNS database. The dataset was split at random into 10 folds, 9 folds being used for training and 1 fold being held-out for testing.

The neural models were trained on GPU-accelerated hardware (with a training time of 6-8 hours) using a batch size of 128 and two passes on the training set. Adam was employed as optimisation algorithm (Kingma and Ba, 2014). The dimension of the embeddings was set to 16, and the maximum length of the domain name was capped to 50 characters. Rectified linear units were used as activation function for the hidden layers.

The evaluation results are provided in Table III. The first row represents a weak baseline using a single feature: the total number of domain queries. This baseline model classifies the record as malicious if its domain has less than 10 queries in the passive DNS dataset, and classifies it as benign otherwise. As we can see, the neural models achieve better results than "shallow" logistic regression models without hidden layers. Furthermore, all feature types (numerical features, rankings, graph-based features, categorical features and domain name) seem to contribute positively to the classification performance. The empirical results are consistent with the findings from Antonakakis et al. (2010) who reported a true positive rate of 97 % given a false positive rate of 0.38 %. These rates are, however, not directly comparable as they are evaluated on different datasets (collected at different time periods).

We can improve the model performance even further by

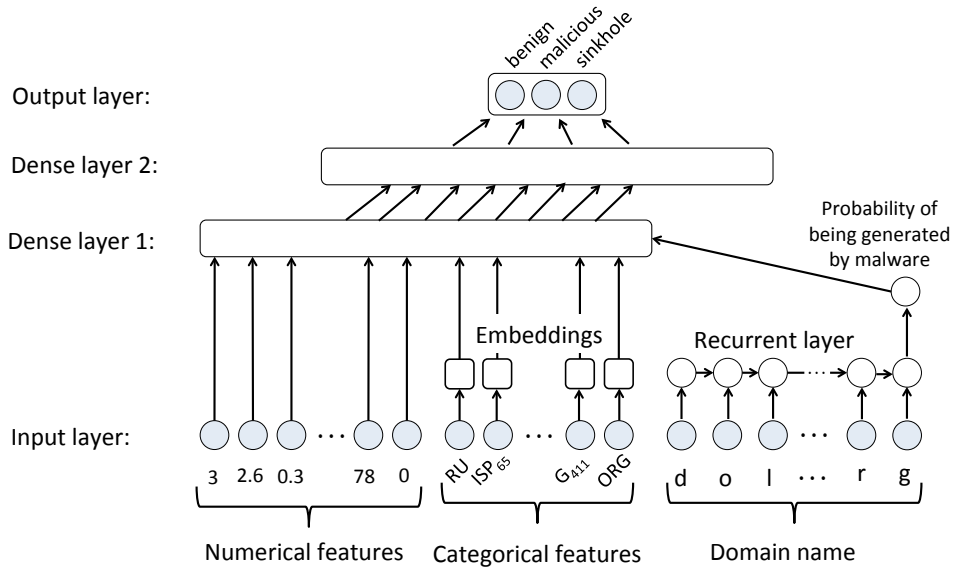

Figure 2: Neural architecture for predicting the reputation of a ⟨domain, IP address⟩ record. The network takes both numerical, categorical and sequential (domain) features as inputs. The categorical features are first converted into lower-dimensional dense embeddings (one per feature). The domain name is fed into a recurrent network (with Gated Recurrent Units) and outputs a prediction on whether the domain is likely to be generated by malware. The numerical features, embeddings and probability value are then fed into two dense feed-forward layers. Finally, the output of the second dense layer is used to produce a probability distribution over the possible reputations.

applying the model in several passes, in a semi-supervised learning fashion (Chapelle et al., 2010). As described in Section III, only one third of the records are annotated with a reputation label. The remaining two thirds are therefore ignored by the reputation models. It is, however, possible to exploit this unlabelled data to yield better predictions. Once a neural model has been trained on the labelled records, it can be applied on the full dataset of 378 million records in order to obtain, for each record, a probability distribution over the three reputation classes. These predicted reputations can be subsequently employed to recompute the values of the graph-based features (i.e. the numbers of neighbouring nodes at distance d that are benign, malicious or sinkholes), and a new neural model can be trained on the basis of these updated features. The classification results of this second model are shown in the last row of Table III, and constitute the best performance on the held-out test set.

As shown in Figure 2, the last layer of the neural model outputs a full probability distribution over possible reputation labels for each input. In other words, the trade-off between recall and false positive rate can be easily adjusted by modifying the threshold with which we decide to classify a record as benign, malicious or sinkhole. This trade-off can be plotted using a ROC curve, shown in Figure 3. One can also interpret the classification performance in terms of recall (also called true positive rate) that can be achieved given a fixed false positive rate. For the best performing model (a neural model with 3 hidden layers each of dimension 1024, all input features and two learning passes), we achieve a recall for the detection of malicious records of 0.74 for a false positive rate of 1:100K, 0.86 for a 1:10K, 0.92 for 1:1000, 0.97 for 1:100 and 0.99 for 1:10.

VII. Conclusion

This paper presented a novel machine learning approach to the problem of predicting the reputation of domain names and IP addresses. The approach relies on a deep neural architecture combining a broad range of features related to the properties and relational structure of the DNS records. The model is trained on a large dataset extracted from passive DNS monitoring. To our knowledge, this paper is the first one to investigate the use of deep neural networks for the development of dynamic reputation models. The evaluation results presented in Section VI demonstrate that the model is able to predict whether a DNS record is benign, malicious or a sinkhole with high accuracy.

Future work will focus on online learning strategies for the reputation models. Passive DNS data and reputation lists are indeed regularly updated, and the model should be able to reflect these changes without needing to retrain its parameters from scratch. Online learning strategies will necessitate both efficient ways of performing graph inference on an continuously evolving graph as well as the use of incremental learning algorithms.

Model	**Inputs**	**Benign**			**Malicious**			**Sinkhole**			Accuracy	AUC
		P	R	F_1	P	R	F_1	P	R	F_1		
nb_domain_queries < 10		0.98	0.44	0.61	0.10	0.87	0.19	0.0	0.0	0.0	0.54	0.603
Logistic regression (no hidden layers)	B	0.96	0.98	0.97	0.65	0.51	0.57	0.00	0.00	0.00	0.945	0.851
Logistic regression (no hidden layers)	B, R	0.96	0.98	0.97	0.65	0.51	0.57	0.00	0.00	0.00	0.945	0.866
Logistic regression (no hidden layers)	B, N	0.94	0.98	0.96	0.95	0.04	0.08	0.02	0.31	0.04	0.916	0.922
Logistic regression (no hidden layers)	B, N, R	0.94	0.98	0.96	0.91	0.05	0.09	0.02	0.27	0.04	0.918	0.925
Logistic regression (no hidden layers)	B, N, C	0.99	0.99	0.99	0.89	0.87	0.88	0.89	0.98	0.93	0.983	0.987
Logistic regression (no hidden layers)	B, D, C, R	0.99	0.99	0.99	0.85	0.81	0.83	0.79	0.94	0.86	0.976	0.977
Logistic regression (no hidden layers)	B, D, N, R	0.97	0.97	0.97	0.60	0.65	0.62	0.51	0.26	0.35	0.944	0.945
Neural model with 1 hidden layer (dim=1024)	B, D, N	0.98	0.99	0.99	0.86	0.79	0.82	0.52	0.82	0.63	0.975	0.976
Neural model with 1 hidden layer (dim=2000)	B, D, N, C, R	1.00	0.99	0.99	0.93	0.94	0.93	0.99	1.00	0.99	0.991	0.997
Neural model with 1 hidden layer (dim=1024)	B, D, N, C	1.00	0.99	0.99	0.90	0.94	0.91	0.98	1.00	0.99	0.988	0.996
Neural model with 1 hidden layer (dim=1024)	B, D, N, C, R	0.99	0.99	0.99	0.93	0.93	0.93	0.99	1.00	0.99	0.990	0.996
Neural model with 1 hidden layer (dim=1024)	B, N, C, R	0.99	0.99	0.99	0.93	0.93	0.93	0.98	1.00	0.99	0.990	0.996
Neural model with 2 hidden layers (dim=2000)	B, D, N, C, R	1.00	0.99	0.99	0.89	0.95	0.92	0.98	1.00	0.99	0.988	0.997
Neural model with 2 hidden layers (dim=1024)	B, D, N, C, R	1.00	0.99	0.99	0.92	0.95	0.93	0.98	1.00	0.99	0.990	0.997
Neural model with 2 hidden layers (dim=1024)	B, D, N, C	1.00	0.99	0.99	0.90	0.95	0.92	0.99	1.00	0.99	0.989	0.997
Neural model with 3 hidden layers (dim=1024)	B, D, N, C	1.00	0.99	0.99	0.89	0.95	0.92	0.99	0.96	0.97	0.988	0.995
Neural model with 3 hidden layers (dim=1024)	B, D, N, C, R	1.00	0.99	0.99	0.90	0.96	0.93	0.99	1.00	0.99	0.989	0.997
Neural model with 4 hidden layers (dim=1024)	B, D, N, C, R	1.00	0.99	0.99	0.91	0.95	0.93	0.97	1.00	**0.99**	0.990	0.997
Neural model with 3 hidden layers (dim=1024) and semi-supervised learning	B, D, N, C, R	1.00	**1.00**	1.00	0.97	0.96	**0.96**	0.99	0.96	0.98	**0.995**	**0.99**

Table III: Evaluation results for the task of classifying DNS records into benign, malicious or sinkhole records. The results are computed from a held-out section of the passive DNS data (10 % of the records with reputation labels, amounting to a total of 13 165 489 records).
The precision (**P**), recall (**R**) and F_1 score are provided for each of the three classes (namely benign, malicious and sinkhole). The precision is the fraction of records classified by the model as belonging to class X that are actually X, while the recall (also called sensitivity or true positive rate) is the fraction of records of class X that are classified as X by the model. Finally, the F_1 score is an harmonic mean of the two measures. The accuracy is simply the ratio of records that are correctly classified. Finally, the AUC (Area Under Curve) is the area under the Receiver Operating Characteristic (ROC) curve associated with the model predictions. The ROC Curve (shown in Figure 3 for a subset of the models) plots the true positive rate (TPR) against the false positive rate (FPR) when the discrimination threshold for the model is varied. An equivalent interpretation of the AUC score is the probability that the model will rank a randomly selected positive instance higher than a randomly selected negative instance.
The second column indicates the inputs provided to the model. **B** denotes the basic numerical features, **R** the features from Alexa rankings, **N** the features extracted from neighbouring nodes, **C** the categorical features, and **D** the domain names. Due to the large number of potential combinations of these features (along with the combinations of models to evaluate), only a relevant subset of combinations were tested in this paper.

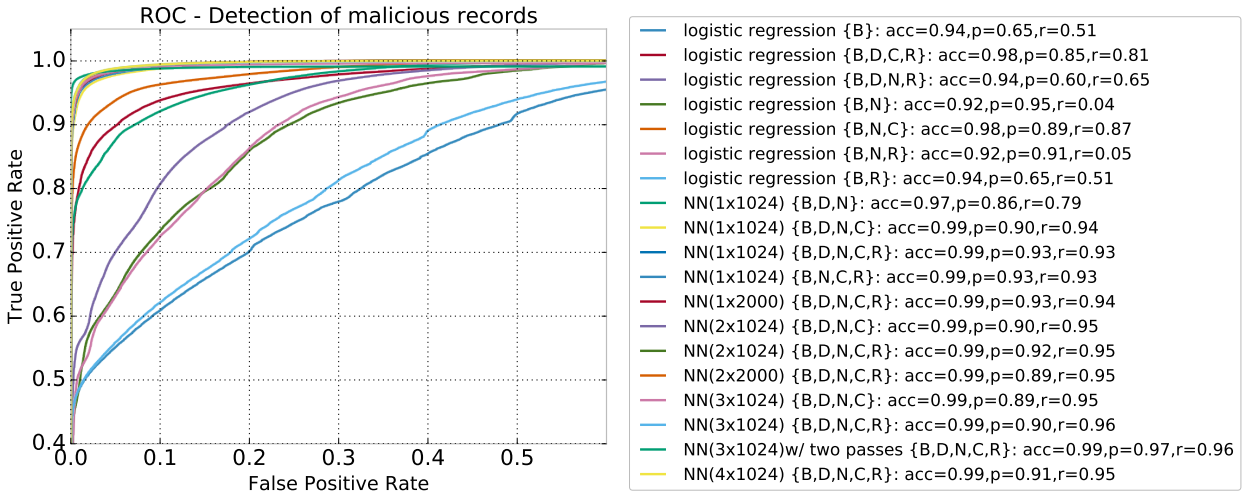

Figure 3: ROC curve for the task of detecting malicious records

ACKNOWLEDGEMENT

The authors would like to thank Mnemonic AS for providing us with their passive DNS database.

REFERENCES

M. Antonakakis, R. Perdisci, D. Dagon, W. Lee, and N. Feamster. Building a dynamic reputation system for DNS. In *Proceedings of the 19th USENIX Security Symposium*, pages 18–18, Berkeley, CA, USA, 2010. USENIX Association.

M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, and D. Dagon. Detecting malware domains at the upper DNS hierarchy. In *Proceedings of the 20th USENIX Security Symposium*, volume 11, pages 1–16. USENIX Association, 2011.

M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. From throw-away traffic to bots: Detecting the rise of DGA-based malware. In *Proceedings of the 21st USENIX Security Symposium*, volume 12. USENIX Association, 2012.

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.

G. Bruneau and R Wanner. DNS sinkhole. *SANS Institute InfoSec Reading Room, Aug*, 7, 2010.

O. Chapelle, B. Scholkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010.

L. Deri, S. Mainardi, M. Martinelli, and E. Gregori. Graph theoretical models of DNS traffic. In *Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1162–1167, 2013.

Y. Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research (JAIR)*, 57:345–420, 2016.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

I. Khalil, T. Yu, and B. Guan. Discovering malicious domains through passive DNS data graph analysis. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 663–674. ACM, 2016.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

P. Lison and V. Mavroeidis. Automatic detection of malware-generated domains with recurrent neural models. *arXiv preprint arXiv:1709.07102*, 2017.

C. Peng, X. Yun, Y. Zhang, S. Li, and J. Xiao. Discovering malicious domains through alias-canonical graph. In *Proceedings of the IEEE Trustcom-BigDataSE-ICESS 2017 Conference*, pages 225–232. IEEE, 2017.

L. Watkins, S. Beck, J. Zook, A. Buczak, J. Chavis, W. H. Robinson, J. A Morales, and S. Mishra. Using semi-supervised machine learning to address the Big Data problem in DNS networks. In *Proceedings of the 7th Annual IEEE Computing and Communication Workshop and Conference (CCWC)*, pages 1–6. IEEE, 2017.

S. Williams, Leonid Oliker, Richard Vuduc, John Shalf, Katherine Yelick, and James Demmel. Optimization of sparse matrix–vector multiplication on emerging multicore platforms. *Parallel Computing*, 35(3):178–194, 2009.

Y. Zhou, Q.-s. Li, Q. Miao, and K. Yim. DGA-based botnet detection using DNS traffic. *Journal of Internet Services and Information Security*, 3(3/4):116–123, 2013.