1 Dataset

1.1 Data set requirements

Network data set for malware detection based on Domain Name System (DNS) are very limited. As noted in [**?**], not all malware behaves in the same way and the choice of malware used to infect a machine is of great importance during the design phase of the dataset. In the design of the [**?**] network data set, the malicious activity has been chosen accordingly to its capacity to generate network traffic, otherwise it has not been included in the data set generation. In addition, since our work is based on malware trying to establish a connection with the DNS server, we have two further requirements:

- the malware must establish a connection with the Command and Control (C&C) server,
- it must use Domain Generation Algorithm (DGA) algorithms. Hence, this requirements reduce the number of data sets compatible with our experiment.

1.2 Data set used

The data set used make use of network traffic capture provided by the Malware Capture Facility Project (MCFP) developed by [**?**], a repository of captured network generated by infected or not-infected machines. The project provides hundreds of captures, diveded by the so-called normal, which means not-infected, and the infected ones. For each capture, we have:

- a Packet CAPture (PCAP) file.
- The malware if the capture is infected.
- Other files generated by network analysis tools like Argus.

Given this repository, we need to check for each capture if it would be compatible with our purpose. The compatibility check consists of analysing for each capture the amount of DNS traffic:

- In the case of a not-infected capture, we can only hope that the amount of traffic is the greater possible.
- For an infected capture, we check if the malware produce DGA traffic.

Therefore, for each capture we performed the following steps:

1. Given the PCAP, we filter out all the packets except the PCAP ones.
2. Checking in this filtered PCAP, the presence of a DNS query which looks like a DGA malware.
3. Given the amount of DNS queries:

- If it is low or the query generated are not DGA, we discard the capture.
- Otherwise we insert each DNS packet of the capture into a relational database.

1.3 Database

In order to achieve better performance during our experiments and analysis, instead of releaving in sparse csv files, we implement a relational database. The principal table of the database are:

- PCAP-table, indicating each @capture. It is related to the Malware-table.
- Malware-table, indicating the malware which infected one or more captures included in the PCAP-table.
- Packet-table, indicating each DNS-packet. Each packet is related to its parent PCAP.
- DN-table (Domain Name table), indicating each domain name apperead in Packet-table, avoiding duplicates. It is related to one or more records of Packet-table.
- NN-table (Neural Network table), indicating each @LSTM neural network used to predict DGA domain name.
- DN-NN-table, a many to many relationship which relates each domain name of DN-table to a neural network of NN-table, including the prediction value $\varepsilon_i = O_i(d_j)$ where i indicate the NN record, and j the the DN record.

Using this methodology, we avoid:

- Duplication of work, the prediction for each packet will be performed just one time for each neural network.
- Duplication of data, the information about the same domain name - and its predictions - will not be duplicated for each time it appears. Further advantages are:
- Saving data store memory.
- Use of SQL language.
- Better data management related to a test-bed made of *sparse* CSV.
- Each capture insterted into the Database follow the same data processing steps and fits into the database data structure.

	count	q	u
infected	33	13.1M	113k
not-infected	17	299k	29k

Table 1: Amount of requests for each class.

	count	q	u
infected	33	396k	3.42k
not-infected	17	17.6k	1.71k

Table 2: Average number of requests and uniques per @capture.

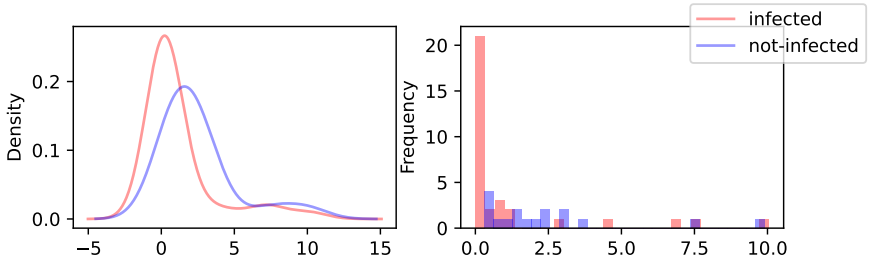

Figure 1: {
Histogram and density distribution of q/s per class.}

	count	mean	std	min	25%	50%	75%	max
infected	33	1.17	2.46	1.87m	13.3m	108m	764m	10.1
not-infected	17	2.49	2.58	310m	838m	1.77	2.91	9.92

Table 3: Average number of queries/uniques per second grouped by the infection class.

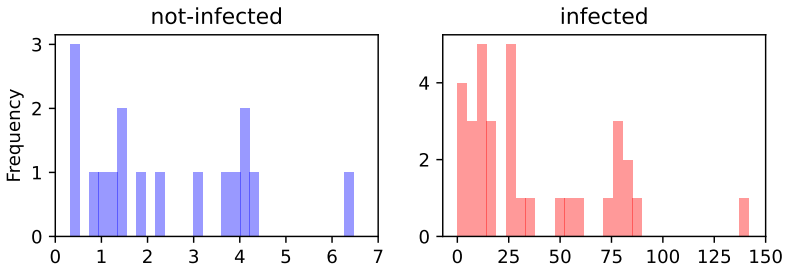

Figure 2: {
Distribution of the duration of all the captures per class.}

dga	mean	std	min	25%	50%	75%	max
infected	18 days	17 days	1.4 min	5.6 days	13 days	30 days	71 days
not-infected	2.4 hr	1.8 hr	19 min	1.1 hr	1.9 hr	3.9 hr	6.5 hr

Table 4: bo

1.4 Data set analysis

The final data set is composed by 50 captures. Of these, 33 are infected and 17 not-infected. Table 1 shows the amount of requests for each class. If we consider the **duration**, we will have the unbalancing ratios showed in Table 3. We can note that:

- the not-infected q/s are more *sparse* relatevely to the infected ones.
- the infected q/s average is lower than the not-infected one.

The problem is that the duration is very different for the two kind of captures.

Captures duration The capture duration is higly unbalanced. As we can see in Table 4 precisely:

- The maximum duration for not-infected is just 6.5 hours respect the 71 days of the infected ones.
- The average duration for not-infected is just 2.4 hours while the infected one is 18 days.