Revealing User Behavior by Analyzing DNS Traffic

Martín Panza, Diego Madariaga, Javier Bustos-Jiménez

Revealing User Behavior by Analyzing DNS Traffic

Martín Panza[1], Diego Madariaga[1], and Javier Bustos-Jiménez[1]

NIC Chile Research Labs, University of Chile
{martin,diego,jbustos}@niclabs.cl

Abstract. The Domain Name System (DNS) is today a fundamental part of Internet's working. Considering that Internet has grown in the last decades as part of human's culture, user patterns regarding their behavior are present in the network data. As a consequence, some of these human behavior patterns are present as well in DNS data. With real data from the '.cl' ccTLD, this work seeks to detect those human patterns by using Machine Learning techniques. As DNS traffic is described by a time series, particular and complex techniques have to be used in order to process the data and extract this information. The procedure that we apply in order to achieve this goal is divided in two stages. The first one consists of using clustering to group DNS domains basing on the similarity between their users' activity. The second stage establishes a comparison between the obtained groups by using Association Rules. Finding human patterns in the data could be of high interest to researchers that analyze the human behavior regarding Internet's usage. The procedure was able to detect some trends and patterns in the data that are discussed along with proper evaluation measures for further comparison.

Keywords: DNS · Clustering · Association Rules · Human Behavior

1 Introduction

As a critical component in Internet's infrastructure, the Domain Name System (DNS) plays a vital role in Internet's working. As the system that translates the domain names to IP addresses, every web service relies on it to operate. For its part, with the continuous growth of users, Internet is nowadays an important element that affects humans' life and culture in an undeniable way. Taking this into consideration, human behavior patterns can be recognized in the Internet's data flow; and as a consequence, in the DNS traffic. These patterns make this source of data highly valuable for the analysis and understanding of the human conduct over the usage activity on Internet.

As an example of this statement, one can identify a strong periodic behavior when simply visualizing the amount of queries in DNS traffic. The periodicity showed in Figure 1 is caused by the high traffic that people generate during the day, and the low traffic at night when the majority of people rest. Likewise, human activity is higher during weekdays rather than during the weekend, as

can also be seen in Figure 1, where Saturday and Sunday correspond to the two lower peaks of the time series. This data corresponds to the Chilean country code top-level domain (ccTLD): '.cl', and it is the data that we will use later for experimentation in this work, with a further description in Section 3.

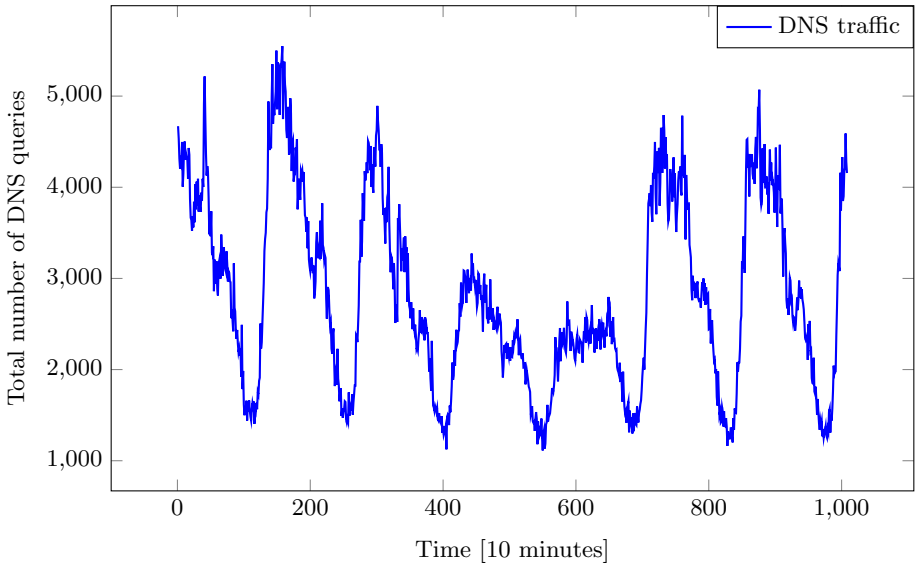

Fig. 1. DNS traffic time series

Moreover, given the purpose of DNS as a way of using IP addresses in a human-understandable way, it contains even more information regarding these behavior patterns. Many times one can easily speculate on the content of a webpage by looking at its domain name.

Recognizing and studying these patterns could be of high relevance for researchers interested in analyzing the human behavior on Internet usage. As well as for resources managing, that DNS operators might be interested in using to improve the service provided by their systems.

This work seeks to use Machine Learning techniques on real DNS traffic from authoritative servers in order to discover and analyze human patterns, showing a useful process for this purpose based on methods and evaluation measures from relevant related work.

Considering that DNS traffic could be described by time series, we apply methods and distance measures specifically designed for this purpose; as time series analysis is a topic of research itself that has acquired huge relevance in the literature in recent years. Mainly due to its applicability on several and diverse topics, for example, financial markets, brain activity, and astronomy.

2 Related Work

The study of human behavior has always been of great interest to researchers, mainly in the field of sociology [2]. However, understanding human behavior in computer science is an emergent research field that has significantly benefited from the rapid proliferation of wireless devices that frequently report status, and location updates [3].

Most of state of the art works that address the study of human behavior through the analysis of networking-related data exploit the high periodicity present in network data. This high periodicity and, consequently, low entropy, is mostly attributed to the impact of the regularity of human patterns [7, 13] on the network state [16].

Recently, the temporal and spatial analysis of data traffic on the mobile network [16] has shown how different human patterns have different effects over the network state, generating distinct patterns of the data traffic in diverse locations. Also, as researches have shown how this periodicity is also present in DNS data [12], important conclusions about user behavior can be deduced at analyzing this portion of the network, in order to optimize the performance of this critical component of the Internet.

Time series analysis has become a very popular topic of research lately. Specially because of its usage on popular topics, such as financial markets; and because concepts like similarity and summarization have many different visions depending on the problem [4]. On top of it, data mining on time series studies have developed various adaptations of the common techniques [6] since, in general, each problem is addressed with an original procedure depending on its conditions.

3 Data-Set Overview

The data-set used in this work consists of a week of normal operation traffic of one of the authoritative DNS servers of the '.cl'. It starts on 7 November, 2018, until 14 November of the same year. '.cl' is the country code top-level domain (ccTLD) of Chile, administrated by NIC Chile. Every DNS packet from queries to the server and responses to users is present in the data-set. The server studied belongs to an anycast configuration along with other servers.

A time series of DNS traffic was built by aggregating all the successful server responses into 10-minutes intervals. Therefore, each point of the time series corresponds to the number of DNS packets from server responses with record types 1, 2, 15, or 28 (A, NS, AAAA, MX) obtained in ten minutes of data. For the purpose of this work, only the most important domains on '.cl' were considered; in view of the vast number of domains that this ccTLD is responsible for, most of which contain low activity. We based on Amazon Alexa's top sites [1] to determine the most relevant domains for our study. We made a further selection based on the number of queries received for those domains, resulting in 82 high activity domains of the Amazon Alexa's top sites. All the time series together manifest a total of 2,854,260 DNS packets.

The Figure 1 in Section 1 shows an aggregation of the whole time series used in this work, showing the total number of DNS queries every 10 minutes during the studied week.

Since the data comes from a normal working of the system, it takes on great importance in the analysis of this work and gives relevance to the results obtained as users patterns are captured in the traffic.

4 Methodology

Considering the domains that were taken into account basing on the criteria described in Section 3, an experimental procedure was made consisting of two stages that are further described in the following sections.

The first stage corresponds to a clustering analysis on all the time series, in order to find groups of domains according to their traffic activity from the number of queries received from the users. Each domain's time series was preprocessed by applying a Simple Moving Average (SMA) method and a Z-Score normalization to them, with the purpose of reducing noise and capturing the regular shape of the time series, as well as reducing the scale, which was convenient for the distance measures that were used. In this way, giving the clustering algorithm a smoother and consistent input. The time series clustering algorithm used in the experiments was the Partitioning Around Medoids (PAM) [10] for multiples values of k. The selected value of k used for further analysis was determined by the internal clustering validation measure: Davies-Bouldin Index [5]. With regard to the time series distance metric used by the algorithm, the Shape-Based Distance proposed by Paparrizos and Gravano [14] was established in a sliding window of 12 hours, i.e. half-a-day. Before the execution of the experiment, different tags were assigned to each domain as a way of both give a description about the domain's content type, and to evaluate the results using an external clustering validation measure: Rand Index. Lastly, after obtaining the results and selecting k, we display the groups given by the algorithm and discuss the nature of their domain members.

The objective in the second stage was to establish a comparison between the groups obtained in the clustering analysis. To achieve this goal, an association rules analysis was made on a representative of each of the groups, corresponding to the centroid from each cluster obtained in the previous stage. The algorithm used in this phase was the Apriori algorithm [9]. However, to properly feed this algorithm with the time series, a previous procedure to transform time series to a set transactions was done. The most relevant rules were showed in the Results section, and later discussed in the Discussion section.

Some important aspects of this process were implemented using the R packages *dtwclust* [15], containing time series clustering tools, and *apriori* [8] for association rules analysis.

Finally, some conclusions and future work are proposed in the final section.

5 Clustering Analysis

5.1 Algorithm and Configuration

The clustering algorithm used for the experiments is the Partitioning Around Medoids, using the distance measure Shape-Based Distance.

Partitioning Around Medoids Partitioning Around Medoids (PAM) is different from k-means algorithm since it uses elements from the data-set as centroids. The advantage is that it is less sensitive to outliers as it minimizes dissimilarities between the clustering members, and not squared euclidean distances as k-means does. It does require a similarity measure.

The algorithm proceeds as follows:

1. Select k domains as medoid-domain.
2. Link all the other domains to their closest medoid-domain.
3. Calculate the total cost (sum of dissimilarities).
4. While (total cost decreases) do:
   - For each medoid-domain do:
     - For each non-medoid-domain do:
       * Use the non-medoid-domain as medoid-domain instead of the current medoid-domain.
       * Link all the other domains to their closest medoid-domain.
       * Recalculate the total cost.
       * If the total cost increased, then undo the substitution between the medoid-domain and the non-medoid-domains.

A specific advantage of this algorithm to the benefit of this work is that, due to its nature, the final centroids are members from one of each cluster. In Section 6 we use this aspect to directly choose candidates for the Association Rules analysis.

Shape-Based Distance The Shape-Based Distance (SBD) is a similarity measure for time series. It is less costly than the popular Dynamic Time Warping (DTW). It is described by the following equation:

$$SBD(\boldsymbol{x}, \boldsymbol{y}) = 1 - \max_{w}\left(\frac{CC_w(\boldsymbol{x}, \boldsymbol{y})}{\sqrt{\|x\| \cdot \|y\|}}\right) \tag{1}$$

where $CC(x, y)$ is the cross-correlation and w is a value that maximizes $CC_w(x, y)$ based on the convenient shift of the time series with regard to the other one.

This measure reaches values between 0 to 2, and it is highly sensitive to scale. That is why a normalization is required. We used Z-Score normalization as suggested by the distance's authors. In addition, we used a half-a-day window size for the calculations of the similarity.

5.2 Evaluation

Clustering validation measures are divided in two types regarding the information that they require: internal and external. Both have the objective of determining how good the clusters obtained by a clustering algorithm are.

While internal validation measures only require spatial information of the clusters themselves, external validation measures use information that instructs how the result is expected to be, such as what cluster members should or should not be together.

Since we are not interested in adjusting the algorithm to obtain a particular result, an internal validation measure was used for the evaluation of the clustering algorithm: Davies-Bouldin measure. More specifically, it was used to compare the quality of the clusters obtained for different values of k (number of clusters).

Nonetheless, tags were still given to each domain as a way of providing a description of what the domains are related to, allowing further discussion, and also allowing an additional external evaluation.

The tags assigned to each domain are showed in Table 1.

Table 1. Descriptive tags assigned to the domains

Tag	Description
BA	Banking
BS	Big Stores
EC	E-Commerce
ED	Educational
GO	Governmental
JS	Job Sites
OS	Online shopping
NP	Newspaper
PD	Postal Delivery
RS	Radio Station
SE	Search Engine
SU	Supermarket
TC	Telecommunication
TO	Tourism
TV	Television

Davies-Bouldin Index Davies-Bouldin Index (D-B) is given by the following equation:

$$DB = \frac{1}{N} \sum_{i=1}^{N} D_i \tag{2}$$

where N is the number of clusters, and:

$$D_i = \underset{i \neq j}{max}(R_{i,j}) \tag{3}$$

$$R_{i,j} = \frac{S_i + S_j}{M_{i,j}} \tag{4}$$

$$S_i = \frac{1}{T_i} \sum_{j=1}^{T_i} d(x, c_i) \tag{5}$$

$$M_{i,j} = d(c_i, c_j) \tag{6}$$

where c_i is the centroid of the cluster i, T_i is the size of the cluster i, and $d(c_i, c_j)$ is the distance between the two clusters.

This index measures the average distance between each cluster and its most similar one. Thus, a lower score means that the quality of the clusters is better.

Rand Index The Rand Index (RI) is a similarity measure between two clustering solutions. It is given by the following equation:

$$RI = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

where TP corresponds to the True Positives, i.e. the number of elements that are grouped together in both clustering results. TN are the True Negatives, elements that are separated in different clusters in both clustering results. FP and FN are False Positives and False Negatives. They represent the elements that belong to the same cluster only in one of the two clustering solutions, but don't belong to the same cluster in the other clustering solution. In which one of the clustering solutions this happens determines what would be a FP or a FN.

In this case, our tags compose a clustering solution that will be compared to the corresponding clustering solution after selecting the k value, in order to obtain the Rand Index.

5.3 Data Pre-processing

With the purpose of reducing the noise in the time series and capturing the essence of their shape to facilitate the establishment of comparisons between the clustering algorithm, a smoothing and normalization process was made on every time series. First, a Simple Moving Average (SMA) was performed with five as the number of periods, in order to reduce noise. Secondly, a Z-Score normalization was applied to modify the scale of the data, as the distance measure to be used is sensitive to scale.

5.4 Experimental Results

The clustering algorithm was performed for different values of k (number of clusters) in the range from 2 to 10. Davies-Bouldin Index was obtained for each execution. Table 2 shows the score for each value of k. As denoted on it, the minimum score was obtained by k=6, which corresponds to the best number of clusters according to the evaluation measure. Therefore, the clustering results for k=6 were considered for the following experiments in this work.

Table 2. Davies-Bouldin Index for number of clusters k.

k	D-B Index
2	0.483
3	0.448
4	0.348
5	0.333
6	**0.292**
7	0.462
8	0.493
9	0.403
10	0.397

Table 3 displays the groups obtained by the clustering algorithm for six different groups, listing all their members by their domain names. It also presents the Rand Index described in 5.2.

As way of visualizing what is contained inside the clusters, Figure 2 shows plots for each cluster with all the time series of the domains that belong to that particular cluster together.

Fig. 2. Clusters members plot

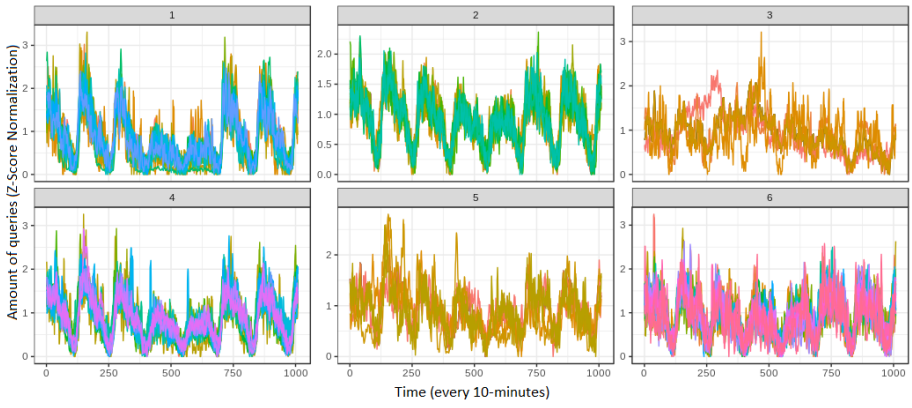

Table 3. Domains and tags by clusters

Cluster	Domain	Tag
1	aiep	ED
	bancoedwards	BA
	bancosantiago	BA
	bci	BA
	bluex	PD
	chileatiende.gob	GO
	chilexpress	PD
	correos	PD
	dt.gob	GO
	entel	TC
	mercadopublico	GO
	officebanking	BA
	scotiabankazul	BA
	scotiabankchile	BA
	scotiabank	BA
	sii	GO
	sistemadeadmision	GO
2	13	TV
	24horas	TV
	adnradio	RS
	biobiochile	RS
	elmostrador	NP
	mega	TV
	paris	BS
	pcfactory	BS
	soychile	NP
	t13	TV
	tvn	TV
	wom	TC
3	airbnb	TO
	google	SE
	redgol	NP
	tripadvisor	TO
4	bancochile	BA
	bancoestado	BA
	bancofalabella	BA
	bancoripley	BA
	claveunica.gob	GO
	cmr	BS
	dafiti	OS
	despegar	TO
	emisora	RS
	lider	SU
	mercadolibre	EC
	pjud	GO
	publimetro	NP
	registrocivil	GO
	ripley	BS
	santander	BA
	sodimac	BS
	trabajando	JS
	transbank	BA
	yapo	EC
5	clarochile	TC
	df	NP
	groupon	TO
	itau	BA
	linio	OS
6	abcdin	BS
	bsale	OS
	buscalibre	OS
	chileautos	EC
	chiletrabajos	JS
	chilevision	TV
	comunidadescolar	ED
	conicyt	ED
	cooperativa	RS
	curriculumnacional	ED
	duoc	ED
	easy	BS
	extranjeria.gob	GO
	inacap	ED
	laborum	JS
	mercadopago	BA
	mineduc	GO
	mitarjetacencosud	BA
	movistar	TC
	santotomas	ED
	uc	ED
	uchile	ED
	udec	ED
	webescuela	ED

Rand Index
0.772

5.5 Discussion

As observed in Figure 2, the process successfully made groups of domains depending on the attributes of each time series. Still in such a straightforward visualization, differences between the arrangements of the time series can be seen between distinct clusters. One clear aspect is on the weekend, that can be

easily identified as the lower peaks in the middle zone of the time series in Cluster 1. These peaks indeed correspond to Saturday and Sunday in the data. Meaning that users of those domains reduce their activity on weekends. On the other hand, members from others clusters, such as Cluster 6 do not clearly demonstrate these distinctions between weekdays and weekends, as users of those domains maintain a uniform usage throughout the whole week. Moreover, members from Cluster 3 show a completely opposite behavior, with peaks on weekends. Nevertheless, all the domains seem to share in common a decrease of activity during nighttime.

The clusters listed in Table 3 also demonstrate a valuable outcome as patterns can be observed when taking into account the content type of the domains, specially when considering our initial descriptive tags. For instance, every domain originally tagged as Educational [ED] was grouped together in Cluster 6, just for *aiep* who was assigned to Cluster 1. This tag considers many of the most important universities and institutes in Chile. Such as Universidad de Chile (*uchile*), Universidad Católica de Chile (*uc*), Universidad de Concepción (*udec*), and Departamento Universitario Obrero y Campesino (*duoc*). As well as some government educational-related domains, such as *conycit* (National Commission for Scientific and Technological Research), and also *mineduc* (Ministry of Education) and *curriculumnacional* (National Curriculum) that were originally tagged as Governmental [GO]. Logically, this kind of domains should present similar traffic, and this is successfully recognized by the algorithm. However, some other not-related domains are also included in the cluster, such as *chilevision* [TV] or *chileautos* [EC].

Another estimable result is the group formed on Cluster 2. As it contains all the domains tagged as Television [TV], except for one. It also incorporates two Radio-Station [RS], and two Newspaper [NP] tagged domains. If we consider that all these tags fit as part of mass media, then we distinguish an interesting pattern captured by our procedure.

We can also observe that the three domains that we manually tagged as Postal-Delivery [PD] were grouped together in the Cluster 1. In this cluster there are also five Governmental [GO] domains and seven Banking [BA] domains.

Additionally, two of the four domains previously tagged as Tourism [TO] were grouped in the smallest cluster along two other domains. One of them is *google* that has a unique tag Search-Engine [SE], expected by us to be distinguished from the rest, assumption that was partially fulfilled.

Given all the above, it is possible to assure that human behavior patterns influence the DNS traffic of the domains, establishing important differences between them, that can be detected by the used time series distance measure. Moreover, these patterns can be detected by the clustering algorithm to successfully create groups whose members show similar behavior and are very likely to share content meaningful to humans. Thus, detecting human patterns in DNS is feasible by employing clustering techniques.

6 Association Rules

In order to establish comparisons between the clusters obtained from the procedure of Section 5, association rules are expected to highlight the trends and patterns within the time series. The resulting centroid from every cluster, which corresponds to a domain's time series, was considered as the representative for the experiments and analysis performed in this section. In this way, the association rules procedure was applied on six time series representing the members of each cluster.

The association rules algorithm used is the popular Apriori algorithm. In order to feed it with our data, some transformations were required as a preprocessing stage. That is why an SAX was used to convert the time series to symbols, in addition to a rule for feature extraction.

6.1 Apriori Algorithm

Apriori algorithm was designed to generate association rules that indicate patterns and trends inside a data-set composed by multiple collections of items, commonly associated with transactions. It focuses on the frequency with which the items appear in the transactions, and with what other items they are usually present.

The algorithm receives a minimum support as input, as well as the transactions, and generates candidate itemsets whose appearances in the transactions are filtered by the minimum support given. Finally, it outputs all the association rules that remain. Selecting the relevant rules after this process falls completely to the user criteria, depending on some common evaluation indicators for these rules:

1. Support:
$$Supp(X) = \frac{|\{t \epsilon T; X \subseteq t\}|}{T} \tag{8}$$
2. Confidence:
$$Conf(X \rightarrow Y) = \frac{Supp(X \cup Y)}{X} \tag{9}$$
3. Lift:
$$Lift(X \rightarrow Y) = \frac{Supp(X \cup Y)}{Supp(X) \times Supp(Y)} \tag{10}$$

where T is the total number of transactions and t is a single transaction.

6.2 Data Pre-Processing

Given that Apriori algorithm receives a list of transactions as input, a transformation is needed to be previously made to the time series. A direct solution is transforming the time series to symbols and pass collections of symbols to

the algorithm. This is taken care of by the Symbolic Aggregate approXimation (SAX) [11].

However, SAX uses Piecewise Aggregate Approximation (PAA) to obtain the symbolic values. This procedure reduced the time series length from 1008 points to 168. Five symbols were used in the transformation, resulting in the following time series:

$$S = \{\, s_t : t \,\epsilon\, T,\; s \,\epsilon\, \{a, b, c, d, e\} \,\} \tag{11}$$

where e corresponds to the highest values of the previous time series, and a to lowest ones. Also, $|T| = 168$.

Additionally, one last feature was added to the time series to maintain some relevant information. Using the remaining time series 12, each symbol was assigned an integer in the following way:

a = 0
b = 1
c = 2
d = 3
e = 4

This with the purpose of obtaining the difference every two points in the time series as a way to establish a measure of flow change in the traffic to not only know its position at a given time, but also its direction.

For example, if a time series has the symbol b at a given point, and in the next point it changes to d, we will note this change as $d - b = 4 - 2 = 2$, and we will say that it increased by 2.

Adding this feature and grouping by every two points leaves our final DNS traffic time series as:

$$S = \{\, (s, n)_t : t \,\epsilon\, T;\; s \,\epsilon\, \{a, b, c, d, e\}\,;\; n \,\epsilon\, \mathbb{N};\; n \,\epsilon\, [-4, 4] \,\} \tag{12}$$

With $|T| = 84$.

This is the final form of the time series that the Apriori algorithm received as a transactions array.

6.3 Results

Table 4 shows what we consider as the most relevant rules after mining the association rules resulting from the Apriori algorithm. The table is subdivided by rules that contain only numeric values, only alphabetic values, and both of them.

Table 4. Relevant Association Rules obtained from the Apriori algorithm

Number	Body	Head	Support	Confidence	Lift
1	C3=0, C4=-1	C5=-1	0.11	0.818	2.864
2	C4=-1, C5=-1, C6=-1	C2=-1	0.06	1	5.600
3	C2=2, C4=2	C1=2	0.04	1	28
4	C2=1	C3=0	0.13	0.917	1.510
5	C3=0, C5=0, C6=0	C4=0	0.18	1	1.615
6	C2=0, C3=-1	C5=0	0.13	0.917	1.878
7	C1=0, C2=0, C3=0, C5=0, C6=0	C4=0	0.10	1	1.615
8	C5=-1, C6=-1	C2=-1	0.07	0.857	4.800
9	C2=-1, C3=0, C5=-1	C6=-1	0.06	0.833	4.118
10	C1=0, C2=-1	C4=-1	0.07	1	4
11	C2=-1, C5=-1	C4=-1	0.10	0.889	3.556
12	C2=-1, C4=-1	C5=-1	0.010	0.800	2.800
13	C2=0, C3=-1, C6=0	C5=0	0.11	1	2.049
14	C1=1, C2=0	C6=0	0.08	1	1.474
15	C1=0, C2=0, C4=1	C6=1	0.036	1	12
16	C1=a, C2=a	C6=a	0.18	1	5.600
17	C1=a, C4=a, C5=a, C6=a	C2=a	0.12	1	5.250
18	C2=c, C3=e	C6=c	0.08	1	4.941
19	C2=b, C3=a, C6=b	C4=b	0.07	1	4.667
20	C1=e, C4=e	C6=e	0.14	0.800	4.200
21	C5=e, C6=e	C1=e	0.13	0.917	4.053
22	C6=e	C4=e	0.18	0.938	3.938
23	C1=e, C4=e, C6=e	C5=e	0.12	0.833	3.684
24	C4=b, C6=b	C2=b	0.13	0.917	3.667
25	C1=c, C6=b	C2=b	0.07	0.857	3.429
26	C1=b, C6=c	C5=b	0.07	0.857	3.130
27	C1=c, C6=b	C2=b	0.07	0.857	3.429
28	C3=c, C6=c	C4=b	0.06	0.833	3.889
29	C1=d, C4=c	C1=-1	0.06	1	3.652
30	C2=c, C5=b	C5=1	0.06	0.833	5
31	C2=0, C5=0, C1=e, C5=e	C3=-1	0.07	0.857	4.500
32	C4=0, C3=a	C3=0	0.08	1	1.647

6.4 Discussion

The rules showed in Table 4 indicate some patterns in the comparison between the members of each cluster obtained in Section 5.

For example, rule number 3 tells us that every time there was a big increase (magnitude 2) experienced in the Clusters 2 and 4, there was also the same increase in Cluster 1 with a tremendously high value of lift. However, not with a big value of support.

Number 4 tells with high support that an increase in Cluster 2 will be likely to be accompanied with no change in Cluster number 3. This sets up some differences between the clusters' behavior that would not be easy to see otherwise.

Rule number 2 states, with a high lift value, that if Clusters 4, 5, and 6 experience a decrease, you can safely expect that Cluster 2 will decrease too. With higher support but lower lift, rule number 8 states that if only Clusters 5 and 6 decrease, Cluster 2 will decrease likewise. Rules number 11 and 12 indicate that this behavior will also occur in the other way. That is, if Cluster 2 experiences a decrease along with 4 or 5, the remaining one will be very likely to decrease as well.

Rule number 3 says that if Cluster 3, 5 and 6 maintain their value, Cluster 4 will maintain its value too. However, rule number 13 says that Cluster 5 will maintain its value when both Cluster 2 and 6 do not change, and Cluster 3 is experiencing a decrease.

As for the rules containing symbols, some rules like 16, 17, 21, and 22 tell us what clusters tend to stay in their peaks or valleys when other clusters experience the same. However, other rules such as number 18 tell us that when some clusters are currently in their top or bottom values, others can be found in their middle values; in this case Cluster 6 always obtained c value when Cluster 2 was in c, but Cluster 3 was in his peak e.

Rule number 28 tells us that when Clusters 3 and 6 stay in their middle values, Cluster 4 is very likely to be lower on activity than them.

Finally, some more complex rules regarding both symbols and numeric changes were obtained in the last rows. For example, they tell us that when Cluster 2 has value c and Cluster 5 has value b, Cluster 5 tends to increase with very high lift index. (Rule number 30).

Another case is in rule number 32, saying that when Cluster 4 is not changing its activity and Cluster 3 is at its lowest activity, Cluster 3 tends to maintain its behavior as well. This corresponds to information that is tremendously hard to obtain by other means.

7 Conclusions and Future Work

The procedure proposed in this work was able to identify some patterns in the used time series data. The first stage of our experimentation was able to group domains that have similar content meaningful for humans, obtaining an acceptable external evaluation index as a way for further comparison, but most importantly demonstrating semantic coherence in the domains that were grouped together. As for the second stage, association rules showed interesting trends when comparing the centroids from each cluster that could be useful for performing further analysis and pattern mining.

Taking these results into account, we conclude that human patterns are present in the DNS data, and that these techniques were able to find some of them. This demonstrates that they could be mined and recognized using the appropriate methods and data processing.

Every step from our procedure was associated with an evaluation index as a way of comparison. We suggest as future work the use of other methods that could both find different patterns in the data, and improve the quality of their

extraction. Moreover, we claim an achievement of our goal of finding human patterns present in DNS data, however we encourage a more in-depth analysis of the patterns singularly, with the purpose of recognizing more detailed information about them. We strongly believe that these patterns could be of interest for researchers that analyze the human behavior, in this case over activity on the Internet.

References

1. Amazon: Amazon alexa topsites (2019), https://www.alexa.com/topsites
2. Berelson, B., Steiner, G.A.: Human behavior: An inventory of scientific findings. (1964)
3. Bui, N., Cesana, M., Hosseini, S.A., Liao, Q., Malanchini, I., Widmer, J.: A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques. IEEE Communications Surveys & Tutorials **19**(3), 1790–1821 (2017)
4. Cassisi, C., Montalto, P., Aliotta, M., Cannata, A., Pulvirenti, A., et al.: Similarity measures and dimensionality reduction techniques for time series data mining. Advances in data mining knowledge discovery and applications pp. 71–96 (2012)
5. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence (2), 224–227 (1979)
6. Fu, T.c.: A review on time series data mining. Engineering Applications of Artificial Intelligence **24**(1), 164–181 (2011)
7. Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L.: Understanding individual human mobility patterns. nature **453**(7196), 779 (2008)
8. Hahsler, M., Chelluboina, S., Hornik, K., Buchta, C.: The arules r-package ecosystem: Analyzing interesting patterns from large transaction datasets. Journal of Machine Learning Research **12**, 1977–1981 (2011), http://jmlr.csail.mit.edu/papers/v12/hahsler11a.html
9. Jiawei, H., Kamber, M., Kaufmann, M.: Data mining: Concepts and techniques. 2001. University of Simon Fraser (2001)
10. Kaufman, L., Rousseeuw, P.J.: Partitioning around medoids (program pam). Finding groups in data: an introduction to cluster analysis **344**, 68–125 (1990)
11. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. pp. 2–11. ACM (2003)
12. Madariaga, D., Panza, M., Bustos-Jiménez, J.: Dns traffic forecasting using deep neural networks. In: International Conference on Machine Learning for Networking. pp. 181–192. Springer (2018)
13. Oliveira, E.M.R., Viana, A.C., Sarraute, C., Brea, J., Alvarez-Hamelin, I.: On the regularity of human mobility. Pervasive and Mobile Computing **33**, 73–90 (2016)
14. Paparrizos, J., Gravano, L.: k-shape: Efficient and accurate clustering of time series. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. pp. 1855–1870. ACM (2015)
15. Sarda-Espinosa, A.: dtwclust: Time Series Clustering Along with Optimizations for the Dynamic Time Warping Distance (2019), https://CRAN.R-project.org/package=dtwclust, r package version 5.5.4

16. Wang, H., Xu, F., Li, Y., Zhang, P., Jin, D.: Understanding mobile traffic patterns of large scale cellular towers in urban environment. In: Proceedings of the 2015 Internet Measurement Conference. pp. 225–238. ACM (2015)