

UNIVERSIDADE DE PASSO FUNDO
Programa de Pós-Graduação em
Computação Aplicada

Dissertação de Mestrado

**IDENTIFICANDO
COMPORTAMENTOS AGRESSIVOS
DE RESOLVEDORES DNS ATRAVÉS
DE APRENDIZADO DE MÁQUINA
NÃO SUPERVISIONADO**

NATÁLIA GOMES KNOB



UNIVERSIDADE DE PASSO FUNDO
INSTITUTO DE CIÊNCIAS EXATAS E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

IDENTIFICANDO COMPORTAMENTOS
AGRESSIVOS DE RESOLVEDORES DNS
ATRAVÉS DE APRENDIZADO DE MÁQUINA
NÃO SUPERVISIONADO

Natália Gomes Knob

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Computação Aplicada na Universidade de Passo Fundo.

Orientador: Prof. Marco Antônio Sandini Trentin

Coorientador: Prof. Ricardo de Oliveira Schmidt

Passo Fundo

2022

CIP – Catalogação na Publicação

K72i Knob, Natália Gomes
Identificando comportamentos agressivos de resolvedores
DNS através de aprendizado de máquina não supervisionado /
Natália Gomes Knob. – 2022.
76 f. : il. ; 30 cm.

Orientador: Prof. Marco Antônio Sandini Trentin.
Coorientador: Prof. Ricardo de Oliveira Schmidt.
Dissertação (Mestre em Computação Aplicada) –
Universidade de Passo Fundo, 2022.

1. Internet - Programas de computador. 2. Aprendizado
do computador. 3. Nomes de domínio na Internet. I. Trentin,
Marco Antônio Sandini, orientador. II. Schmidt, Ricardo de
Oliveira, coorientador. III. Título.

CDU: 004.738.5

ATA DE DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO DO ACADÊMICO

NATÁLIA GOMES KNOB

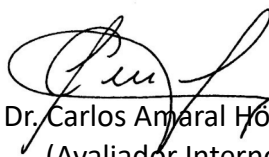
Aos 11 dias do mês de março do ano de dois mil e vinte e dois, às oito horas, realizou-se, de forma on-line, por meio de videoconferência, a sessão pública de defesa do Trabalho de Conclusão de Curso “Identificando comportamentos agressivos de resolvedores DNS através de aprendizado de máquina não supervisionado”, de autoria de Natália Gomes Knob, acadêmica do Curso de Mestrado em Computação Aplicada do Programa de Pós-Graduação em Computação Aplicada – PPGCA. Segundo as informações prestadas pelo Conselho de Pós-Graduação e constantes nos arquivos da Secretaria do PPGCA, a aluna preencheu os requisitos necessários para submeter seu trabalho à avaliação. A banca examinadora foi composta pelos doutores Marco Antônio Sandini Trentin, Ricardo de Oliveira Schimdt, Carlos Amaral Hölbig e Weverton Luis da Costa Cordeiro. Concluídos os trabalhos de apresentação e arguição, a banca examinadora considerou a candidata **APROVADA**. Foi concedido o prazo de até quarenta e cinco (45) dias, conforme Regimento do PPGCA, para a acadêmica apresentar ao Conselho de Pós-Graduação o trabalho em sua redação definitiva, a fim de que sejam feitos os encaminhamentos necessários à emissão do Diploma de Mestre em Computação Aplicada. Para constar, foi lavrada a presente ata, que vai assinada pelos membros da banca examinadora e pela Coordenação do PPGCA.



Prof. Dr. Marco Antônio Sandini Trentin – UPF
Presidente da Banca Examinadora
(Orientador)



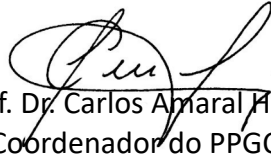
Prof. Dr. Ricardo de Oliveira Schimdt – UPF
(Coorientador)



Prof. Dr. Carlos Amaral Hölbig – UPF
(Avaliador Interno)



Prof. Dr. Weverton Luis da Costa Cordeiro – UFRGS
(Avaliador Externo)



Prof. Dr. Carlos Amaral Hölbig
Coordenador do PPGCA

AGRADECIMENTOS

Agradeço a minha família por todo suporte que sempre me foi dado durante o período do mestrado e pelos ensinamentos, ao longo da vida, de como o estudo nos é essencial. Agradeço principalmente ao meu parceiro de todas as horas, Luís, pela paciência, incentivo e compreensão neste período tão importante de estudo.

Agradeço ao DNS-OARC pela disponibilização dos dados utilizados para o desenvolvimento do meu estudo e pela disponibilização de ambiente onde os pesquisadores podem desenvolver suas atividades. Sem tudo isso, esse trabalho não seria viável.

Agradeço à Universidade de Passo Fundo pela oportunidade de elevar meus conhecimentos e pelo aprendizado prestado através dos professores com quem tive a oportunidade de trocar experiências, ideias e entendimentos. Em especial, agradeço aos professores Dr. Marco Antônio Sandini Trentin e Dr. Ricardo de Oliveira Schmidt pela dedicação, parceria e ensinamentos a mim dispensados.

IDENTIFICANDO COMPORTAMENTOS AGRESSIVOS DE RESOLVEDORES DNS ATRAVÉS DE APRENDIZADO DE MÁQUINA NÃO SUPERVISIONADO

RESUMO

O Sistema de Nomes de Domínio (DNS) é um componente fundamental na infraestrutura da Internet. Ele permite que os usuários facilmente acessem recursos de Web sites usando nomes memorizáveis e humanamente reconhecíveis. O bom funcionamento do DNS exige o uso extensivo de *cache* a fim de reduzir a latência, melhorar a resiliência do sistema frente a ataques DDoS e reduzir o tráfego na Internet. No entanto, o DNS tem sido repetidamente abusado por resolvedores DNS recursivos, os quais geram consultas excessivas à infraestrutura do sistema e fazem uso indevido de recursos valiosos de servidores de nomes autoritativos, possivelmente sem qualquer propósito útil. É nesse sentido que este trabalho apresenta um método que objetiva classificar, quantificar e identificar quem são e como se caracterizam estes resolvedores DNS com comportamento abusivo, através da utilização de algoritmo de aprendizado não supervisionado, que realiza a clusterização dos resolvedores de acordo com o seu comportamento. Esta pesquisa se utilizou do algoritmo *Gaussian Mixture Models* para realizar o agrupamento dos resolvedores de acordo com os atributos adotados. Os resultados obtidos nessa dissertação demonstram que foi possível identificar quatro grupos de resolvedores com diferentes características e quantidades de integrantes. Para os 5 anos dos *datasets* DITL analisados, foi possível concluir que os resolvedores com maior agressividade e que, portanto, merecem maior atenção e destaque, correspondem a uma faixa de 2,19 a 3,18% do total de resolvedores e foram responsáveis por 85,62 a 94,35% do total de consultas recebidas pelos servidores raiz.

Palavras-Chave: Aprendizagem de máquina não supervisionada, *Day in the Life of the Internet*, *Domain Name System*, *Gaussian Mixture Models*, Resolvedores Recursivos, Servidores DNS raiz.

IDENTIFYING AGGRESSIVE DNS RESOLVER BEHAVIORS USING UNSUPERVISED MACHINE LEARNING

ABSTRACT

The Domain Name System (DNS) is a fundamental component of the Internet infrastructure. It allows users to easily access website resources using memorable and human-readable names. The proper functioning of DNS requires extensive use of caching in order to reduce latency, improve system resilience against DDoS attacks, and reduce Internet traffic. However, the DNS has been repeatedly abused for recursive DNS resolvers that generate excessive queries to the DNS infrastructure misuse precious resources of authoritative nameservers, possibly for no useful purpose. This work presents a method that allows classifying, quantifying and identifying who and how these DNS resolvers with abusive behavior are characterized through the use of an unsupervised learning algorithm that performs the clustering of resolvers according to their behavior. This research used the Gaussian Mixture Models algorithm to group the recursives according to the chosen attributes. The results obtained in this dissertation demonstrate that it was possible to identify four groups of recursive resolvers with different characteristics and amounts of components. For the 5 years of the analyzed DITL datasets, it was possible to conclude that the most aggressive recursives, which therefore deserve more attention and study, correspond to 2.19 up to 3.18% of the total resolvers and were responsible for 85.62 up to 94.35% of the total queries received by root servers.

Keywords: Day in the Life of the Internet, Domain Name System, Gaussian Mixture Models, Recursive resolvers, Root Servers, Unsupervised Machine Learning.

LISTA DE FIGURAS

1	Estrutura em níveis do sistema DNS.	17
2	Fluxo de resolução de domínio.	19
2	Consultas recebidas pelos <i>root servers</i> em diferentes anos divididas por TLD e tipo de recurso requisitado.	38
3	Requisições acumuladas de acordo com resolvedores únicos ao longo dos diferentes DITL.	41
4	Distribuição quantitativa das consultas dos 30 resolvedores com maior número de requisições para o ano de 2016	43
5	Distribuição quantitativa das consultas dos 30 resolvedores com maior número de requisições para o ano de 2017	43
6	Distribuição quantitativa das consultas dos 30 resolvedores com maior número de requisições para o ano de 2018	44
7	Distribuição quantitativa das consultas dos 30 resolvedores com maior número de requisições para o ano de 2019	44
8	Distribuição quantitativa das consultas dos 30 resolvedores com maior número de requisições para o ano de 2020	44
9	Exemplo de resolvedores com diferentes características de envio de consultas presentes no DITL 2019	45
10	Exemplos de resolvedores e picos identificados no DITL 2019.	51
11	Distribuição de frequências de resolvedores por atributos no DITL 2019.	51
12	Correlação entre variáveis através do método Spearman.	52
13	DITL 2019 agrupado por meio de K-Means.	53
14	Clusterização para DITL de 2016.	56
15	Clusterização para DITL de 2017.	57
16	Clusterização para DITL de 2018.	58
17	Clusterização para DITL de 2019.	59
18	Clusterização para DITL de 2020.	60
19	Comparação da quantidade percentual de resolvedores e consultas presentes em cada grupo, pós classificação, para cada DITL.	62
20	Top resolvedores agressivos existentes em comum entre diferentes anos de DITL.	65
21	Comparativo entre classificações usando todo o conjunto e parte do conjunto de dados de 2016.	69

LISTA DE TABELAS

1	Servidores DNS raiz.	18
2	Tipos de registros.	20
3	Estrutura de diretório de <i>datasets</i> do projeto DITL.	28
4	Informações gerais dos <i>datasets</i> DITL utilizados.	34
5	Informações da coleta de dados nos servidores de nomes DNS.	35
6	Estatísticas das requisições recebidas agrupadas por tipo de recurso requisitado.	39
7	Estatísticas das requisições recebidas agrupadas por TLD requisitado.	39
8	Estatísticas das requisições recebidas agrupadas por letra de servidor raiz que recebeu a requisição.	39
9	TOP 10 resolvedores IP referente aos DITL de 2016, 2017, 2018, 2019 e 2020.	42
10	Informações da coleta de dados nos servidores de nomes DNS.	61
11	Medianas dos atributos para cada grupo classificado e DITL.	62
12	Informações sobre resolvedores agressivos.	63
13	Top resolvedores agressivos comuns para cada ano de DITL.	64
14	Quantidades de top resolvedores agressivos comuns entre diferentes anos de DITL.	64
15	Top resolvedores agressivos ordenados por quantidade de consultas realizadas.	66
16	Estatísticas para o DITL 2016.	69

LISTA DE SIGLAS

AS – Autonomous System

BGP – Border Gateway Protocol

DDOS – Distributed Denial of Service

DITL – Day in the Life of the Internet

DNS – Domain Name System

DNSSEC – Domain Name System Security Extensions

GMM – Gaussian Mixture Models

IDS – Intrusion Detection System

IP – Internet Protocol

ISP – Internet Service Provider

K-NN – K Nearest Neighbor

OARC – Operations, Analysis, and Research Center

RFC – Request for Comment

SIDN – Stichting Internet Domeinregistratie Nederland

SVM – Support Vector Machine

TLD – Top-level Domain

TTL – Time to Live

SUMÁRIO

1	INTRODUÇÃO	13
2	REVISÃO DE LITERATURA	16
2.1	DNS	16
2.1.1	Definições e hierarquia	16
2.1.2	<i>Resource records</i>	20
2.1.3	<i>Cache e TTL</i>	20
2.1.4	<i>Anycast e balanceamento de carga</i>	22
2.1.5	DNSSEC	23
2.2	APRENDIZADO DE MÁQUINA	23
2.2.1	Aprendizado supervisionado	24
2.2.2	Aprendizado não supervisionado	25
2.2.2.1	<i>Gaussian Mixture Model</i>	26
2.3	DNS-OARC E PROJETO DITL	27
2.4	TRABALHOS RELACIONADOS	29
3	PROCEDIMENTOS METODOLÓGICOS	33
3.1	<i>DATASETS</i> DITL	33
3.1.1	Limitações na captura e processamento dos dados	35
3.2	DISTRIBUIÇÃO INICIAL DOS DADOS	36
3.3	DISTRIBUIÇÃO DAS CONSULTAS POR RESOLVEDORES	40
3.4	TOP 30 RESOLVEDORES COM MAIOR QUANTIDADE DE REQUISIÇÕES	43
3.5	PREPARAÇÃO DOS DADOS PARA CLASSIFICAÇÃO	45
3.6	ATRIBUTOS PARA CLASSIFICAÇÃO DOS RESOLVEDORES	48
3.7	CLASSIFICAÇÃO DOS RESOLVEDORES ATRAVÉS DE GMM	53
4	APRESENTAÇÃO E DISCUSSÃO DE RESULTADOS	55

4.1	RESULTADOS DA CLASSIFICAÇÃO A PARTIR DO USO DO GMM	55
4.1.1	DITL 2016	55
4.1.2	DITL 2017	57
4.1.3	DITL 2018	58
4.1.4	DITL 2019	59
4.1.5	DITL 2020	59
4.2	ANÁLISE DOS RESULTADOS OBTIDOS	61
4.2.1	Análise dos resolvedores com agressividade alta	63
5	CONCLUSÃO	67
5.1	LIMITAÇÕES E DISCUSSÕES	67
5.2	DIREÇÕES FUTURAS	70
	REFERÊNCIAS	71

1. INTRODUÇÃO

O sistema de nomes de domínio (DNS) pode ser considerado a base da Internet. Isso porque o acesso a sites, envio de e-mail e uso de outras aplicações dependem do seu prévio funcionamento. Cabe a ele a tradução de nomes humanamente reconhecíveis para endereços IP correspondentes, exigidos por todos os softwares de rede. Com isso, o acesso a qualquer recurso é muito mais fácil a qualquer usuário, independentemente do seu conhecimento. Assim, o DNS é usado para fornecer e manter este mapeamento entre domínios e endereços IP, sendo previsto principalmente através das RFCs 1034 e 1035.

Os dados são armazenados em bancos de dados distribuídos onde cada servidor de nomes é responsável (autoritativo) por sua própria parte da árvore de nomes. A delegação de autoridade ocorre por meio de registros NS (servidor de nomes) e deve ser consistente entre os nós pais e os filhos na árvore de nomenclatura [1]. O grau mais elevado desta árvore corresponde aos servidores raiz (*root servers*) sendo que a integridade e a disponibilidade de muitas formas de comunicação na Internet dependem das respostas deles.

Atualmente, são treze os servidores raiz responsáveis por realizar a tradução de endereços por todo o mundo. Eles possuem diversas réplicas e estão dispersos em vários pontos do planeta em um sistema redundante e altamente disponível. Seus arquivos de zona são consultados por resolvedores recursivos os quais correspondem a subsistema utilizado por programas do usuário para encontrar a tradução de um nome para um endereço IP acessando a servidores de resolução de nomes [2]. A cada consulta realizada, os resolvedores devem guardar em *cache* o tipo de recurso solicitado e a validade deste de tal forma que não seja necessário realizar uma nova consulta que solicite a mesma informação enquanto estiver válida a resposta recebida previamente.

Este tempo de validade, conhecido como *time to live* ou tempo de vida em tradução livre é o que determina por quanto tempo uma informação pode ser armazenada em *cache*. E a utilização de cache pelos resolvedores é imprescindível para o bom funcionamento do DNS haja vista que ele reduz a latência em consultas, aumenta a confiabilidade e disponibilidade do DNS e, em alguns casos, até oferece resiliência à ataques DDoS [3].

A crescente dependência da internet para o desenvolvimento de atividades econômicas, educacionais, governamentais, de comunicação e de lazer, e sua presença cada vez mais regular em nossa realidade, traz muitos desafios no que diz respeito ao seu gerenciamento, disponibilidade, segurança e integridade desse sistema. A robustez e a redundância dos protocolos DNS escondem erros de configuração, tentativas de ataques e comportamentos agressivos que, embora individualmente ou em pequeno volume não sejam capazes de desencadear uma indisponibilidade ou dano significativo, estão frequentemente presen-

tes, afetando a segurança de clientes e consumindo recursos imprescindíveis para o correto funcionamento do DNS.

A comunidade vem estudando há algum tempo o ecossistema DNS e analisando seu tráfego. As pesquisas nessa área objetivam responder aos mais diversos tipos de problemas que afetam o sistema, como resiliência a ataques, desempenho, topologia e outros. Embora estudos anteriores tenham investigado servidores DNS [4, 5, 6, 7, 8] e alguns usos de padrões (*patterns*) em grandes detalhes [9, 10, 11, 12, 13], não existe uma definição sobre o que são resolvedores agressivos, e muito pouco se conhece a respeito da diversidade de implementações e comportamentos dos resolvedores recursivos. Contudo, eles são a base de toda a infraestrutura DNS e seu mau comportamento pode afetar os clientes DNS além dos servidores de autoridade.

É notório o conhecimento acerca da existência desses resolvedores com comportamento abusivo, os quais, contrariamente aos preceitos das RFCs, enviam muito mais consultas do que deveria ser tolerado para cada resolvedor. Ainda que, como dito, o DNS seja robusto e redundante, servidores raiz podem ser afetados de diferentes formas e precisam estar preparados para lidar com uma carga muito superior de consultas se comparado a um cenário ideal, escalando seus recursos para fazer frente a tantas requisições. Resolvedores recursivos mal configurados podem ser utilizados indevidamente em ataques [14], ou introduzir atrasos nas respostas aos clientes [9, 15]. Além disso, resolvedores recursivos que geram consultas excessivas à infraestrutura DNS [16, 17, 18] usam recursos preciosos de servidores de nomes autoritativos, provavelmente sem que haja qualquer propósito útil.

Em virtude dessa ausência de entendimento e definição ou mesmo quantificação desses resolvedores, tem-se o seguinte problema de pesquisa: como realizar a identificação, quantificação e classificação de resolvedores DNS com comportamento agressivo?

A fim de dar resposta ao questionamento levantado, o presente trabalho tem como objetivo geral o desenvolvimento de uma metodologia que permita definir e identificar quem são e como se caracterizam estes resolvedores DNS com comportamento abusivos através da utilização de algoritmo de aprendizado não supervisionado. Neste sentido, foram utilizados *datasets* de um dia de tráfego recebido por servidores raiz para cada ano, durante cinco anos, que retratam adequadamente o cenário do DNS, possibilitando a clusterização de resolvedores de acordo com o seu comportamento.

Já como objetivos específicos, necessários para o desenvolvimento deste trabalho, tem-se os que seguem:

- Extração e análise dos *datasets* DITL presentes em servidor DNS-OARC a partir de 2016 e até 2020;
- Definição de atributos através dos quais será realizada a classificação do comportamento dos resolvedores;

- Escolha de algoritmo para aprendizagem de máquina em conformidade com cenário encontrado e característica dos dados;
- Identificação de resolvedores cujo comportamento pode ser entendido como abusivo de acordo com o resultado da clusterização;
- Quantificação de resolvedores a fim de se dimensionar a quantidade de resolvedores que realizam volumes abusivos de consultas aos servidores raiz, precisando-se o quanto, em relação ao total, eles correspondem;
- Validação dos resultados obtidos através da clusterização realizada.

A par de tais objetivos, esta pesquisa constitui-se através da seguinte organização: No capítulo 2 é realizada a revisão de literatura com apresentação de conceitos-chave relacionados ao DNS e aprendizado de máquina, assim como apresentado o projeto DITL e também os trabalhos relacionados ao tema. Em capítulo 3 são tratados detalhes acerca da metodologia desenvolvida para a identificação, quantificação e classificação de resolvedores com comportamento agressivo. Já no capítulo 4 são apresentados os resultados obtivos a partir da clusterização dos dados de *dataset* a partir de algoritmo de aprendizagem não supervisionada e, por fim, em capítulo 5 são apresentadas limitações ao trabalho, direções futuras, assim como realizadas discussões finais acerca dos resultados obtidos.

2. REVISÃO DE LITERATURA

Neste capítulo serão apresentados fundamentos e conceitos gerais que permeiam essa dissertação. Primeiramente, serão abordadas definições e elementos do sistema DNS. Após, serão apresentadas as definições e elementos relacionados à clusterização, uma técnica de *machine learning* que permite a classificação de dados a partir de aprendizado não supervisionado. Por fim, será apresentada a organização e projeto responsável pelos *datasets* utilizados, os quais propiciaram a análise e classificação dos resolvedores recursivos.

2.1 DNS

O *Domain Name System* (DNS) ou, sistema de nome de domínios, consiste em um banco de dados distribuído cujo objetivo é mapear nomes de *host* para endereços IP. Ele permite que os usuários acessem um recurso específico na internet usando diretamente o nome de domínio traduzido por um resolvedor que armazena o resultado dessas consultas em *cache* a fim de reduzir a latência e melhorar a resiliência do sistema. Com isso, o DNS desempenha um papel fundamental na Internet de hoje, permitindo inclusive que sites possam distribuir de forma transparente a carga dos clientes entre servidores da Web replicados ou redirecionar solicitações de clientes de seus próprios servidores para redes de entrega de conteúdo (CDNs), ao realizar associações de nome para endereço dinamicamente e fornecendo associações diferentes para clientes diferentes [19].

Tendo em vista o papel crucial desempenhado pelo DNS, a complexidade em torno do sistema aumentou com o passar dos anos. Se anteriormente tratava-se de ser uma simples ocorrência de consulta de dispositivos finais (*host*) a algum resolvedor DNS que, por sua vez, passaria a consultar servidores de nomes autorizados em nome dos clientes, agora envolve, muitas vezes, camadas de resolvedores e estruturas complexas que fazem requisições através desta infraestrutura agora complicada e por vezes oculta, tornando difícil o entendimento do todo e a atribuição de responsabilidade por comportamentos diferentes observados de diferentes entidades participantes.

A fim de melhor compreender essa estrutura, cabe apresentar suas definições e elementos, os quais constam nas seções a seguir.

2.1.1 Definições e hierarquia

Apesar de conceituado e definido inicialmente em 1983 através das RFC 882 e RFC 883, o DNS foi realmente implementado e consolidado em 1987 por meio das RFC

1034 e RFC 1035, as quais determinaram que ele seria um sistema distribuído, haja vista o tamanho do banco de dados a ser mantido bem como a frequência das atualizações e consultas. Definiu-se, ainda, que adotaria estrutura organizacional hierárquica, com nomes utilizando “.” como o caractere para marcar o limite entre hierarquia de níveis [2].

Com isso, proveu-se um sistema capaz de realizar o mapeamento entre nomes de domínios e endereços IP através de consultas a dados previamente armazenados em um banco de dados distribuído e estruturado em forma de árvore invertida, onde cada nó corresponderia a um domínio. Nesta hierarquia, o mais alto nó da hierarquia é conhecido como domínio raiz e é seguido por outros níveis intermediários que se tornam raízes para novos subdomínios sucessivamente [20].

O DNS corresponde a uma estrutura em níveis, sendo que cada servidor de nome, ou *name server*, mantém informações sobre uma zona específica através de um arquivo que contém os mapeamentos entre nomes de domínio e endereços IP sobre os quais o servidor possui autoridade. Essas zonas são automaticamente distribuídas a servidores de nomes, fornecendo serviço redundante para os dados em uma determinada zona [2].

Na Figura 1, encontra-se representada a estrutura em níveis do sistema DNS, sendo possível observar que os *root servers* ocupam a posição mais alta entre os níveis.

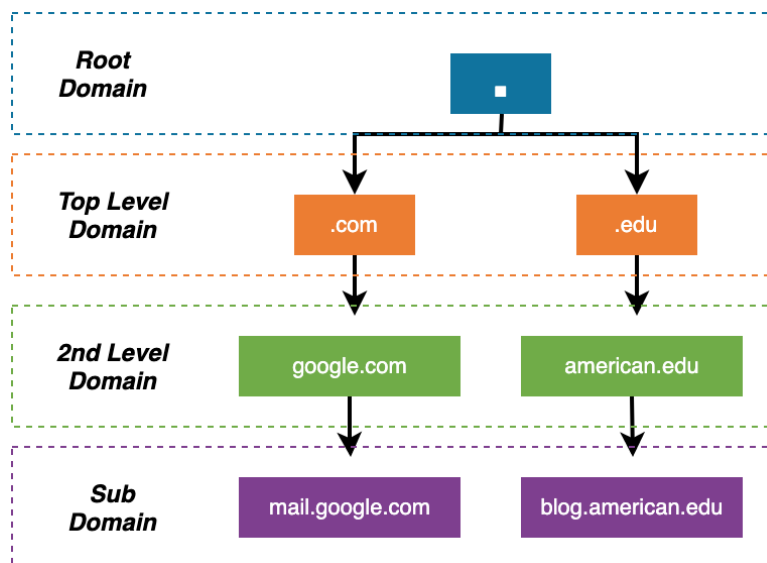


Figura 1: Estrutura em níveis do sistema DNS. Fonte: Adaptado de CHOWDHURY, C. [21].

Servidores raiz, ou *root servers*, correspondem ao ponto de partida para consultas que objetivam resolver nomes para endereços IP. Atualmente existem 13 *root servers* em operação, os quais são compostos por réplicas em *anycast* que respondem a solicitações feitas por *resolvers*, ou resolvidores, retornando o endereço do servidor com autoridade no domínio de nível superior [22].

Esses 13 servidores raiz são mantidos por 12 diferentes operadores, cada um identificado por uma letra, conforme a Tabela 1. Eles operam em *anycast*¹ através de suas múltiplas instâncias, a fim de fornecer um serviço confiável mesmo em caso de falha de *hardware* ou *software*.

Tabela 1: Servidores DNS raiz. Fonte: Root Servers Org. [22]

Servidor	Operador	Qtde Locais/Instâncias
A-root	Verisign, Inc.	16/53
B-root	University of Southern California/ISI	6/6
C-root	Cogent Communications	10/10
D-root	University of Maryland	149/156
E-root	NASA (Ames Research Center)	254/308
F-root	Internet Systems Consortium, Inc.	242/267
G-root	US Department of Defense (NIC)	6/6
H-root	US Army (Research Lab)	8/8
I-root	Netnod	64/72
J-root	Verisign, Inc.	118/185
K-root	RIPE NCC	73/79
L-root	ICANN	147/167
M-root	WIDE Project	5/9

Conforme referido anteriormente, os *root servers* recebem consultas realizadas por resolvedores. Na RFC 1034 o termo resolvedor, ou *resolver*, foi utilizado de forma genérica para designar qualquer subsistema utilizado por programas do usuário para encontrar a tradução de um nome para um endereço IP, acessando a servidores de resolução de nomes [2].

Na prática, resolvedores podem corresponder ao sistema que realiza rotinas sem grandes complexidades, procurando em arquivos estáticos a correspondência entre um endereço IP e o nome requisitado, chamados de *stub resolvers*, quanto a resolvedores recursivos fornecidos pelos *Internet Service Providers* (ISPs) ou mesmo a servidores de resolução de nomes (*Resolving Name Server*), os quais, através de consultas iterativas, extraem informações de servidores de nomes a fim de responder às requisições de clientes [24].

Dito isso, considerando-se como exemplo uma consulta por “www.example.com”, ao deixar de encontrar um domínio em seus registros, o resolvedor de sistema fará uma chamada ao servidor de resolução de nomes que, por sua vez, acionará um servidor raiz.

¹Um grupo *anycast* é definido como um conjunto de instâncias que são executadas pela mesma organização e usam o mesmo endereço IP, ou seja, o endereço de serviço, mas são nós fisicamente diferentes. Cada instância anuncia (por meio do sistema de roteamento) a acessibilidade para o mesmo prefixo/comprimento que cobre o endereço de serviço e tem a mesma origem Sistema Autônomo (AS). As instâncias podem empregar uma política de roteamento global ou local. As instâncias locais tentam limitar sua área de captação a seus pares imediatos. As instâncias globais não fazem tal restrição, permitindo que o BGP sozinho determine seu escopo global, mas use o prefixo em seu caminho AS para diminuir a probabilidade de sua seleção em uma instância local [23].

Este, quando pesquisar por “www.example.com” em seu arquivo de zona, não encontrará o parâmetro em seus registros, contudo, fornecerá ao servidor de resolução de nomes o endereço do servidor responsável pelos “com” endereços. Ou seja, servidores raiz não detêm a informação de onde o domínio está hospedado, mas eles direcionam o resolvidor para o servidor de nome específico que opera no domínio de nível superior solicitado conforme poderá ser observado na Figura 2.

Estes domínios de nível superior ou *top-level domains (TDL)*, correspondem à parte mais genérica do domínio, estando à direita do nome e separados por um ponto. Como exemplo, têm-se os domínios EDU, COM, NET, ORG, GOV, MIL e INT, além dos domínios de códigos de países ou *country code top-level domain (ccTLD)* que incluem extensões como FR, NL, KR e US, e são organizados por um administrador para cada respectivo país que pode delegar o gerenciamento de partes da árvore de nomes [25].

Em continuidade ao caso hipotético de consulta do endereço IP para o nome “www.example.com”, o TDL requisitado procuraria em seu arquivo de zona uma entrada correspondente ao nome solicitado e não o encontraria, entretanto, encontraria o endereço do servidor responsável por “example.com”, o que permitiria ao servidor de resolução de nomes realizar uma nova consulta, agora para o servidor de nome de domínio ou *Domain-level name server*, e encontrar, por fim, o endereço IP correspondente ao nome de domínio desejado.

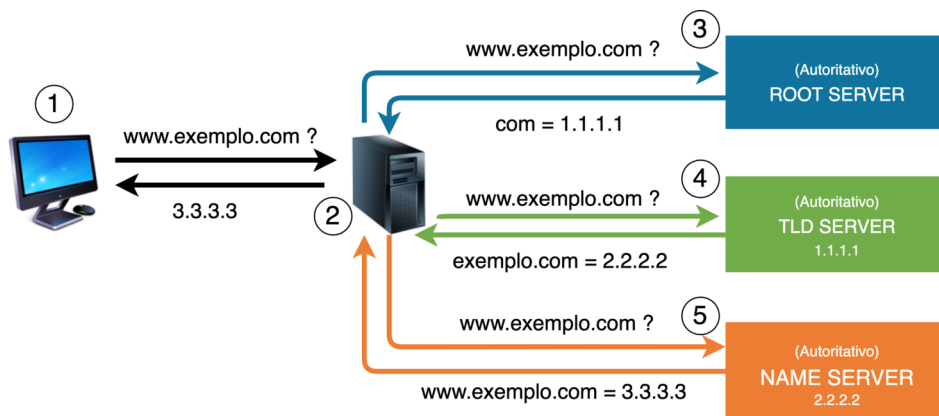


Figura 2: Resolução DNS do domínio www.example.com. Fonte: Adaptado de LIU, B. et al. [26].

Na Figura 2 está representada a dinâmica entre os diversos componentes que estão envolvidos na tradução de nomes em endereços. O servidor *Stub* (1), ao não encontrar em seus registros a tradução para o nome solicitado, consultará o servidor recursivo (2) que, por sua vez, ao não possuir em cache a resposta para a solicitação, acionará sucessivamente um *root domain* (3), um *top-level domain server* (4) e um *name server* (5), os quais responderão autoritativamente à consulta aos domínios constantes em suas zonas

de controle. Com isso, o cliente conseguirá acessar o domínio diretamente através de seu endereço IP (6).

2.1.2 *Resource records*

Cada servidor de nome de domínio possui um conjunto de informações composto por diferentes registros de recursos ou *resource records*. Estes, podem ser entendidos simplesmente como um mapeamento único entre um recurso e um nome. Assim, recursos podem mapear um nome de domínio para um endereço IP, definir os servidores de nomes para o domínio, definir os servidores de correio para o domínio entre outras opções [2].

As transações entre um *host* e um servidor DNS ocorrem com uso de algum dos diversos tipos de registros. Quando o *host* envia um registro “consulta” o servidor DNS responde com um registro “resposta”. Uma parte do registro de recurso contém o cabeçalho, o qual é composto pelo tipo de registro (se é uma consulta ou resposta), código de operação entre outras informações, e a outra parte é composta pelo tipo de registro, comprimento do registro e, em seguida, o conteúdo real do registro [2].

São vários os tipos de registros dentre eles, alguns dos mais comuns e relacionados com o estudo são os A, AAAA, NS e DS cujas utilizações encontram-se na Tabela 2.

Tabela 2: Tipos de registros. Fonte: Adaptado de IANA [27].

Tipo	Id.	Função
A	1	Mapeia um nome de domínio para um endereço IPv4 de 32 bits
AAAA	28	Mapeia um nome de domínio para um endereço IPv6 de 128 bits
NS	2	Delega um subdomínio a um conjunto de servidores de nomes
DS	43	Identifica a chave de assinatura DNSSEC de uma zona delegada

2.1.3 *Cache e TTL*

Resolvedores recursivos caracterizam-se por armazenar o resultado de consultas previamente realizadas. Esse registro temporário denominado *cache* permite que futuras resoluções de um mesmo nome já conhecido possam ser executadas rapidamente, reduzindo a latência, diminuindo a carga do serviço DNS e reduzindo o tráfego na Internet. A duração pela qual o registro é mantido depende do parâmetro *time to live* (TTL) definido no registro do servidor DNS autoritativo. O TTL é expresso em segundos e indica um tempo mínimo pelo qual o registro deverá ser mantido antes de ser descartado. Por exemplo, um TTL “86400” indica que sua validade será de 24 horas [28]. Passado o tempo indicado no TTL, o servidor recursivo deverá fazer uma nova requisição à cadeia de servidores autoritativos para obter uma versão atualizada do registro de recurso.

Cada registro de recurso possui um valor TTL que é definido pelo operador do domínio de DNS. Cabe dizer que não há um consenso definitivo sobre valores corretos de TTL a serem utilizados em diferentes níveis de servidores autoritativos e de acordo com diferentes tipos de registros. Contudo, segundo a RFC 1033, uma boa prática consiste em utilizar valores TTL mais altos e os diminuir a medida em que alguma alteração seja necessária. De acordo com isso, recomenda-se a utilização de valores entre um dia e uma semana, diminuindo-se para uma hora ou um dia até que as alterações sejam implantadas [28]. Isso visa a garantir que *caches* DNS não armazenem registros desatualizados por muito tempo.

Na prática, vê-se que os valores TTL variam muito, embora existam faixas de valores comumente mais adotadas. Em estudo que objetivava descobrir e recomendar valores de TTL mais adequados[29], procurou-se descobrir quais eram os valores de TTL usualmente utilizados na internet. Para tanto foram utilizadas cinco fontes de dados: o ccTLD .nl, listas de ranqueamento dos sites Alexa, Majestic, e Umbrella, e a zona raiz do DNS. Para cada domínio foram solicitados registros de tipo NS, A, AAAA, MX, DNSKEY dos servidores autoritativos de zona filha. Para os recursos de tipo NS e A, notou-se que os valores de TTLs continham grande variação entre si (de 1 minuto a 48 horas) independentemente da origem. Observou-se também que os domínios de nível superior da zona raiz são muito mais conservadores do que as listas Alexa, Majestic, Umbrella, adotando valores correspondente a 24 horas ou superiores. Ainda, viu-se que registros NS tendem a ter TTLs mais longos do que registros do tipo A e que CDNs² utilizam TTLs mais curtos, principalmente para possibilitar o balanceamento de carga.

Valores de TTL poderão ser mais altos ou baixos dependendo da volatilidade das informações. Valores de TTL baixos fazem com que servidores recursivos façam consultas mais frequentes a servidores DNS autoritativos, o que acarreta aumento de carga no servidor mas permite, por outro lado, que as alterações em registros DNS se propaguem mais rapidamente. Por sua vez, valores mais altos diminuem a carga em servidores e garantem maior resiliência no caso de um ataque de negação de serviço, mas podem fazer com que as informações se tornem obsoletas à medida que ocorrem novas alterações [21].

A escolha de um valor TTL adequado é de grande importância, permitindo que informações armazenadas no cache sejam igualmente robustas às informações contidas na origem, diminuindo e equilibrando a carga do fluxo de tráfego dos servidores. De acordo com a RFC 1033, o menor valor de TTL é de um dia, embora seja sabido que é possível utilizar zero, caso em que o registro não será mantido em *cache*, devendo ser consultado toda vez que necessário [30].

²CDN ou *Content Delivery Network* corresponde a um sistema de servidores globalmente distribuídos, instalados em múltiplos *datacenters* cujo objetivo é disponibilizar conteúdo de maneira mais rápida a usuários finais ao diminuir a latência, fornecendo alta disponibilidade e alto desempenho.

Desta forma operadores possuem uma tarefa desafiadora: a de equilibrar valores como segurança, latência e execução de alterações de registros levando em consideração as interações entre o DNS e a internet como um todo.

2.1.4 **Anycast e balanceamento de carga**

Tendo em vista a essencialidade do DNS e o constante aumento de tráfego na Internet, práticas têm sido utilizadas para torná-lo mais eficiente em termos de performance, escalabilidade e disponibilidade. De acordo com isso, passou a ser comum a adoção de soluções como balanceamento de carga (*load balance*) através do *anycast*.

O balanceamento de carga tradicional refere-se à distribuição do tráfego da Internet entre vários servidores que fornecem o mesmo serviço respondendo, contudo, através de diferentes IPs. Nessa estrutura, consultas DNS são respondidas de acordo com fatores tais como a integridade de um servidor, a geolocalização do cliente e até mesmo a prioridade atribuída ao endereço IP. Por sua vez, o *anycast* permite que vários servidores respondam às requisições através de um único endereço IP, atribuindo a solicitação ao servidor mais próximo com capacidade para processar a requisição mais eficientemente [31].

O *Anycast* é amplamente utilizado atualmente, revelando-se indispensável na replicação do serviço DNS. Ele é utilizado por muitos operadores de grandes domínios, tais como *root servers*, *top level domains*, grandes companhias e resolvedores públicos. Em *anycast*, um único endereço IP é anunciado por muitos locais físicos (*sites anycast*), cada um com um ou vários servidores. Para tanto, utiliza-se a política de roteamento BGP para associar clientes (resolvedores recursivos) ao servidor ou grupo de servidores que esteja mais próximo [32].

É comum combinar o uso de *anycast* e balanceamento de carga como em *root servers*, por exemplo. Neste caso, a raiz de DNS é implementada através de 13 diferentes servidores raiz identificados através de letras (A-M), sendo que cada local físico (*site anycast*) pode operar através de múltiplos servidores atrás de um balanceador de carga [33].

Infere-se que a utilização de *anycast* traz uma variedade de vantagens como diminuição da latência em consultas, aumento da confiabilidade e da disponibilidade do DNS e resiliência frente a ataques DDoS, embora em alguns casos indisponibilidades perduram por maior tempo em virtude do tempo de convergência do roteamento BGP [3].

2.1.5 DNSSEC

Visando adicionar mais critérios de segurança ao DNS, desenvolveu-se o *Domain Name System Security Extensions* (DNSSEC). Trata-se de um conjunto de extensões ao DNS que fornecem autenticação e integridade de informações para resolvedores e aplicativos através do uso de assinaturas digitais criptografadas [34]. Por prover autenticação e integridade das informações, o DNSSEC reduz o risco de uso de dados DNS falsificados ou manipulados através de envenenamento de cache DNS (*DNS cache poisoning*) ou mesmo de *DNS spoofing*, por exemplo.

Para tanto, utiliza-se esquemas de assinaturas digitais baseadas em criptografia de chave pública. Assim, a ideia geral é que cada nó na árvore DNS está associado a uma chave pública e que cada mensagem dos servidores DNS é assinado sob a chave privada correspondente. Com isso, cria-se uma cadeia de confiança que garante a autenticidade das delegações de uma zona até um ponto de confiança [35].

Para alcançar seu objetivo, o DNSSEC introduziu vários outros tipos de registros de recursos (RR) com funções específicas, dentre eles o já citado DS que indica qual a chave usada na zona delegada, o RRSIG que contém a assinatura DNSSEC para um conjunto de registros, o NSEC que garante a inexistência de um nome e tipo de registro como parte da validação DNSSEC e o DNSKEY que corresponde a chave pública, incluída na própria zona e que valida as assinaturas digitais de um determinado domínio [36].

2.2 APRENDIZADO DE MÁQUINA

O aprendizado de máquina, considerado uma das áreas aplicadas da inteligência artificial, já existe há praticamente duas décadas, embora apenas mais recentemente tenha se tornado amplamente disponível em virtude do avanço do poder de computação e o armazenamento de dados. Esta técnica se concentra no desenvolvimento de algoritmos que podem aprender com os dados, detectando padrões, de forma a realizar previsões através dos mesmos [37]. O conceito mais conhecido de aprendizado de máquina é, sem dúvida, o atribuído à Arthur Samuel, que define que ele corresponde ao campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados [38].

Algoritmos de aprendizado de máquina podem ser separados em três classes: supervisionados, não supervisionados e por reforço. Algoritmos supervisionados são aqueles cujo aprendizado é guiado através de resultados pré-definidos que servirão para que o modelo aprenda quais deverão ser os seus resultados de saída. O aprendizado não supervisionado se caracteriza pela ausência de exemplos já conhecidos ou de rótulos, de tal forma que o algoritmo deverá realizar a tarefa de classificação sem qualquer *feedback* ex-

terno humano. Por fim, o aprendizado por reforço é baseado no objetivo de maximização de recompensas, as quais fornecidas conforme ações executadas pelo algoritmo [37].

Como neste trabalho são utilizadas técnicas de clusterização e classificação e, portanto, algoritmos de aprendizado não supervisionado e supervisionado, faz-se necessário apresentar alguns conceitos e conhecimentos pertinentes aos temas.

2.2.1 Aprendizado supervisionado

Conforme já referido anteriormente, o aprendizado supervisionado se caracteriza pelo uso de conjuntos de dados rotulados para treinar algoritmos que classificam dados ou predizem resultados e assim produzir uma saída desejada. Desta forma, um modelo é treinado até que possa detectar os padrões e relacionamentos implícitos entre os dados de entrada e os rótulos de saída, permitindo que ele produza resultados de rotulagem adequados para dados até então não classificados [39].

Este tipo de aprendizagem é utilizado para solucionar problemas que envolvem regressão (com os quais se fazem projeções, por exemplo) e classificação (quanto se atribuem, aos dados de teste, categorias específicas) sendo que para a resolução desses últimos, os algoritmos mais conhecidos são *k*-NN, *Naive Bayes*, *Support Vector Machine*, *Decision Trees* e *Logistic Regression* [39].

Talvez o mais simples dos algoritmos de classificação citados acima seja o *k*-Nearest Neighbors(*k*-NN), o qual é considerado não paramétrico, pois a estrutura do modelo é determinada pelo *dataset* utilizado e preguiçoso, pois os dados de treinamento não são aprendidos previamente, mas sim guardados em memória, razão pela qual seu desempenho não é satisfatório quando o número de entradas é muito grande. O *k*-NN classifica os dados com base em uma medida de similaridade, ou seja, funções de distância, sendo o 'k' a variável que determina a quantidade de 'vizinhos' mais próximos a serem considerados quando da verificação da classe que ocorre com maior frequência entre eles [40].

Outro algoritmo muito usado é o *Naive Bayes* o qual é baseado no teorema de Bayes e cujo objetivo é calcular a probabilidade de algo ser 'a' dado 'b'. Este método trabalha com uma suposição 'ingênua' de independência entre as variáveis do problema, e é muito utilizado em tarefas de categorização de texto [40].

Por sua vez, *decision trees* ou árvores de decisão são um tipo de aprendizado de máquina supervisionado em que os itens são classificados ao passarem pelas ramificações de seu nó raiz, em um processo binário. Simplificadamente, pode-se dizer que cada nó interno representa um teste de um atributo, os ramos representam o resultado deste teste e cada nó folha representa uma categoria predefinida. Assim, o resultado é encontrado ao se calcular todos os atributos pelos quais o fluxo passou [39].

O Support Vector Machine (SVM) é derivado de um único classificador linear, denominado *Perceptron*, e se destina à solução de problemas de classificação não lineares. Esse classificador linear define um hiperplano ótimo como o limite entre as classes. Assim, este algoritmo usa uma técnica chamada truque de *kernel* para transformar seus dados e, com base nessas transformações, encontra um limite ideal entre as saídas possíveis [39].

Por fim, regressão logística é um algoritmo de classificação usado para realizar previsões em dados a partir de um conjunto prévio de observações. Suas previsões são baseadas em probabilidades, calculadas a partir de um modelo ajustado, restritas ao intervalo de valores 0-1. Devido ao fato de que um modelo logístico pode ser usado para determinar a probabilidade de a resposta ocorrer com base em valores de predição específicos, chamados de padrões de covariáveis, é frequentemente utilizado pela comunidade estatística nas últimas décadas [41].

2.2.2 Aprendizado não supervisionado

Contrariamente ao aprendizado supervisionado, o aprendizado não supervisionado é aquele que não faz uso de exemplos em que a resposta ao que se quer classificar já é conhecida. Desta forma, o algoritmo, através do emprego de alguma técnica, deverá encontrar padrões ou estruturas a partir dos dados fornecidos para alcançar o objetivo final que será a classificação daquelas entradas.

Para a realização do aprendizado não supervisionado são utilizadas algumas técnicas sendo a principal delas a clusterização, através da qual é realizado agrupamento de dados de acordo com suas similaridades ou diferenças. De forma simplificada, um algoritmo de agrupamento calcula a distância entre os agrupamentos e divide os pontos de dados em vários grupos com base na distância relacional entre eles [37].

É possível afirmar que o algoritmo de clusterização mais conhecido e popular é o K-means. Ele corresponde a uma ferramenta de clusterização que agrupa dados dependendo dos vetores médios de agrupamentos, ou seja, ele procura segregar os dados mais próximos dos centróides. Para tanto, previamente deve-se definir o número de grupos ou *clusters*, representado pelo parâmetro 'K'. Definido o número de grupos a serem formados, a primeira etapa do algoritmo é inicializar centróides, um para cada grupo. Cada item é atribuído ao *cluster* cujo centróide esteja mais próximo. Após, os centróides são atualizados de acordo com a média de valores do *cluster* formado. O arranjo e a atualização dos centróides são iterados até suas convergências ou até que seja atingido o critério de parada [39].

O K-means não é, contudo, o único algoritmo existente para clusterização. Pode-se citar diversos outros a exemplo dos *Affinity propagation*, *Mean-shift*, *Spectral clustering*, *Ward hierarchical clustering*, *Agglomerative clustering*, DBSCAN, *Gaussian mixture model*

e BIRCH, cada um com uma diferente abordagem para o desafio de definir grupos a partir dos dados [40].

Para a execução da pesquisa, optou-se primeiramente pela utilização do K-means e, após vários testes de agrupamento e estudos de características de outros algoritmos, adotou-se finalmente pela utilização do *Gaussian Mixture Model*. Informações relativas a resultados obtidos e problemas verificados na utilização de outras estratégias de clusterização serão apresentados em capítulo 4.

2.2.2.1 *Gaussian Mixture Model*

Enquanto que o K-means é um modelo baseado em distância, o *Gaussian Mixture Model*, ou GMM, é baseado em probabilidade. Ele pode ser definido como uma metodologia de aprendizagem não supervisionada que objetiva descobrir classes em dados e pode ser representado como uma soma ponderada de N densidades de componentes, conforme a fórmula:

$$p(x|\lambda) = \sum_{i=1}^N w_i g(x|\mu_i, \Sigma_i) \quad (1)$$

Onde x é um vetor aleatório D -dimensional, $g(x|\mu_i, \Sigma_i); i, \dots, N$ são as densidades dos componentes e $w = (w_1, \dots, w_N)$ são os pesos da mistura.

Os pesos da mistura devem satisfazer a condição $\sum_{i=1}^N w_i = 1$, e cada um dos componentes é uma função gaussiana multivariante conforme descrito na função:

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \left\{ -\frac{1}{2} (x-\mu_i)' \Sigma_i^{-1} (x-\mu_i) \right\} \quad (2)$$

Um modelo de mistura gaussiana completo é parametrizado por média, variância e peso das misturas de todos os componentes de densidade e pode ser representado pela notação $\lambda = \{w, \mu, \Sigma\}$ [42]. Em regra, utiliza-se o algoritmo expectativa-maximização (EM) para treinamento. Este algoritmo é usado para aplicar iterativamente a estimativa de máxima verossimilhança nos parâmetros do modelo. O critério de máxima verossimilhança tenta encontrar os parâmetros, o que maximiza a probabilidade para um conjunto e treinamento com N pontos de dados [43]. A probabilidade GMM é normalmente descrita como:

$$p(X|\lambda) = \prod_{j=1}^N p(x_j|\lambda) \quad (3)$$

Uma maximização direta não é possível em virtude da não linearidade da verossimilhança. Com isso o algoritmo EM começa com um modelo inicial $\bar{\lambda}$ e estima um novo modelo se a probabilidade de $(X|\lambda)$ é maior ou igual a $(X|\bar{\lambda})$, portanto, $p(X|\bar{\lambda}) \geq p(X|\lambda)$. A

iteração do algoritmo é dividida em duas etapas: expectativa e maximização. Durante a etapa de expectativa, são calculadas as probabilidades de geração de um dado x_i por um componente i . Essas probabilidades são usadas para calcular a quantia de pontos de dados atribuídos a um componente por $n_i = \sum_{j=1}^N p_{ji}$. Em seguida, na etapa de maximização, ajustam-se os parâmetros para maximizar a probabilidade dos dados para essas atribuições, ou seja:

$$\mu_i = \frac{\sum_{j=1}^N p_{ij} x_j}{n_i} \quad (4)$$

$$\Sigma_i = \frac{\sum_{j=1}^N p_{ij} (x_j - \mu_i)(x_j - \mu_i)^T}{n_i} \quad (5)$$

$$w_i = \frac{n_i}{N} \quad (6)$$

A repetição entre essas duas etapas é realizada até que haja a convergência para um ótimo local [44].

Cabe dizer que, enquanto o K-means utiliza função de distância euclidiana para definir clusters a partir dos dados, e que para um resultado satisfatório é necessário que os dados preferencialmente tenham uma distribuição circular em torno aos centróides, o GMM utiliza probabilidade para atribuição dos pontos de dados para grupos, o que funciona bem, mesmo para distribuições de dados que não sejam lineares.

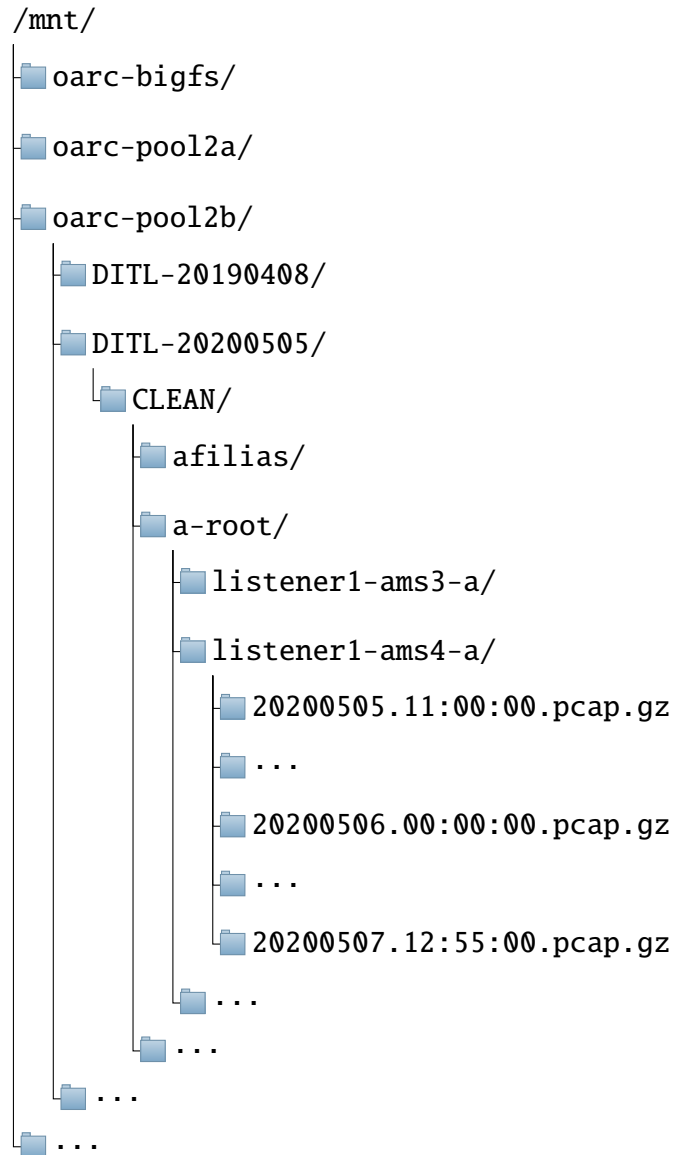
2.3 DNS-OARC E PROJETO DITL

O *Domain Name System Operations, Analysis, and Research Center* ou DNS-OARC corresponde a uma organização sem fins lucrativos que desempenha o papel de aprimorar a segurança e a estabilidade do DNS, assim como a compreensão sobre sua infraestrutura. Esse centro desempenha um papel determinante ao reunir esforços de operadores, implementadores e pesquisadores em uma plataforma através da qual é possível coordenar respostas a ataques, compartilhar informações e conhecimentos a respeito do DNS. Ele é responsável por uma série de atividades, tais como a realização de *workshops*, desenvolvimento de ferramentas e serviços e também análise de ataques sofridos pelo sistema de tradução de nomes. Além disso, a cada ano (normalmente em abril), o DNS-OARC coleta dados da Internet em um determinado dia como parte do projeto denominado *Day In The Life of the Internet* (DITL) [45].

Desde 2006 o DNS-OARC coleta, por horas contínuas e em todos os anos, pacotes DNS nas instâncias dos *root servers* ou servidores raiz. Os dados são mantidos em servidores cujo acesso pode ser realizado através de SSH, possibilitando aos pesquisadores a realização de seus estudos e análises de dados. Por razões de segurança e privacidade, os dados não podem ser retirados dos servidores da OARC. [46]. As coletas que compõem o

DITL geralmente cobrem períodos de 48 horas e são compactados em arquivos pcap, cada qual contendo 5 minutos de captura para cada instância do *root server* presente naquele ano de coleta. A Tabela 3 traz uma demonstração da forma como é a estrutura de árvore do diretório onde residem os dados disponibilizados aos pesquisadores.

Tabela 3: Estrutura de diretório de datasets do projeto DITL. Fonte: Autora.



Conforme se pode observar, o diretório */mnt/* onde são armazenados os dados é composto por vários *pools*, ou seja, sub-diretórios que, por sua vez, armazenam o conjunto de dados referentes a DITL de diversos anos assim como outros dados provenientes de outros projetos e recursos disponibilizados pela DNS-OARC. Os diretórios referentes a cada DITL contêm os dados "crus" (*RAW/*) que estão dispostos tais como enviados por quem os coletou e também os dados processados (*CLEAN/*), já sem inconsistências, que estão prontos para serem utilizados pelos pesquisadores e aos quais efetivamente é permitido o acesso. O diretório *CLEAN/* contém os dados coletados das diversas letras de servidores participantes para aquele DITL e de outros participantes que não são *root servers*. Por

fim, cada diretório correlato a uma letra de servidor raiz é composto por subdiretórios que correspondem às réplicas daquela letra de servidor e armazenam, dentro de si, os arquivos pcaps de todo o período de coleta.

É importante ressaltar que nem todas as instâncias de cada servidor raiz participam a cada ano do projeto DITL, no entanto, ao longo dos anos, cada vez mais há mais réplicas participantes de tal forma que o projeto conta atualmente com conjunto de dados muito mais completos em comparação ao seu início.

2.4 TRABALHOS RELACIONADOS

Ao longo da pesquisa bibliográfica, foram identificados vários trabalhos que possuem estudos relacionados com a manutenção de *cache* DNS e com poluição de tráfego que chega aos *root servers*. Da mesma forma, foram encontrados trabalhos que procuram realizar a detecção de anomalias em tráfego DNS através de aprendizagem de máquina, e outros que utilizam dados do projeto DITL, os quais cabe citar.

No que se refere à utilização de TTL e estratégias de *cache*, Moura destaca que valores de TTL recomendados a servidores recursivos, frequentemente não são atendidos, o que impacta no desempenho do sistema no caso de um ataque de negação de serviço (*Denial of service attack*), podendo, ainda agravar a situação geral. Isso porque, se em um cenário em que os *caches* estão cheios e em que as interrupções do servidor duram menos do que a vida útil do *cache*, cerca de metade dos clientes continuam a receber serviço (ainda que se observe latência de cauda e o aumento no tráfego legítimo), em um cenário em que os *caches* não estão alimentados, a latência e tráfego observadas ficam ainda maiores devido às repetidas tentativas feitas pelos resolvedores [47].

Em trabalho realizado por Schomp no qual se procurou avaliar como diferentes atores tratavam as configurações de tempo de vida (TTL) fornecidas pelos servidores de nomes autoritativos para definir o comportamento dos *caches* de DNS, descobriu-se que em muitos casos o TTL era distorcido antes de atingir o cliente solicitante original, sendo que apenas 19% de todos os resolvedores abertos retornaram valores TTL corretos para as sondas utilizadas [19].

Pang, em outro trabalho relatou uma ampla utilização de registros vencidos por clientes finais. A partir de observações de comportamento específico de vários conjuntos de dados diferentes, notou-se que alguns resolvedores recursivos locais não respeitaram valores de TTL fornecidos por servidores autoritativos e continuaram a utilizar registros já expirados [48].

Em outro viés Moura *et al.*, defendeu que, mesmo quando o valor de TTL a ser usado em cacheamento corresponde ao valor de fato informado pelo resolvedor autoritativo, verifica-se a existência de alguns fatores que podem reduzir a vida útil do *cache* na prática:

caches de resolvedores são de tamanho limitados, os *caches* podem ser liberados prematuramente e os grandes resolvedores podem ter *caches* fragmentados. Quanto ao primeiro caso, pode-se citar o Unbound³ que possui por padrão 4MB de limite para *cache*, valor esse que, contudo, pode ser modificado. Quanto ao segundo caso, os *caches* podem ser descartados antecipadamente, seja a pedido do operador de *cache*, seja acidentalmente ou na reinicialização do software ou da máquina que esteja realizando o cacheamento. Por fim, como resolvedores recursivos tendem a lidar com altas taxas de requisições, podem corresponder a uma estrutura distribuída, atrás de um balanceador de carga ou IP *anycast*. Neste caso, *caches* podem estar fragmentados entre as diversas máquinas que operam individualmente seus *caches* ou podem compartilhar um *cache* de nomes comuns. Essa fragmentação pode reduzir a taxa de *hit* do *cache* [47].

Em trabalho que visou examinar o desempenho do sistema DNS de Danzig *et al.* [49] identificou-se vários erros e avaliou-se o impacto destes nas implementações de DNS. Concluiu-se que a maior parte do tráfego DNS é causada por falhas e configuração incorreta. Levando-se em consideração a eficácia do *cache* de nomes DNS e do cálculo do tempo limite de retransmissão, mostrou-se como algoritmos para aumentar a resiliência levaram a um comportamento desastroso quando os servidores falharam ou quando certas falhas de implementação foram acionadas. Tendo em vista a grande quantidade de tráfego com falha por dia que pode ser gerado a partir dessas falhas, recomendou-se que fossem verificadas ativamente implementações com defeito.

O estudo realizado por Jung *et al.* [50] também analisou o comportamento de pesquisa, coletando traços de área local de DNS para avaliar o desempenho e o comportamento de *cache*. Verificou-se que, embora a maioria das respostas bem-sucedidas fossem recebidas em no máximo duas a três retransmissões, as falhas causam um número muito maior de retransmissões e, portanto, de pacotes que atravessam a rede de longa distância (*Wide area network*). Assim como Danzig, concluiu que as implementações do servidor DNS permanecem excessivamente persistentes, mesmo diante das falhas observadas em ambos os estudos.

No que tange a tráfego DNS ilegítimo, estudo realizado por Brownlee *et al.*[1], estimou que 60-85% das consultas observadas foram repetidas no *host*, oriundas da mesma fonte, dentro do intervalo de medição. Além disso, mais de 14% da carga de consultas no servidor raiz estudado era proveniente de consultas que violavam as especificações DNS. Verificou-se que alguns dos erros anteriormente relatados persistiam [49], além de requisições malformadas, consultas impossíveis e erro relacionado à associação da nomenclatura interna da Microsoft com nomenclatura DNS.

Já em Castro *et al.* [17] analisou-se dados DITL dos anos de 2006 a 2008 a fim de se verificar a quantidade de tráfego DNS ilegítimo que chegava aos servidores raízes.

³Trata-se de um resolvedor de DNS que realiza validação, recursão e cacheamento de respostas desenvolvido pela NLnet Labs. É distribuído gratuitamente em formato de código aberto sob a licença BSD.

Apurou-se que cerca de 98% do tráfego nos servidores é poluição e que, como não há um caminho fácil para se livrar de todas essas consultas indesejadas, operadores de servidores raiz, e aqueles que os financiam, deveriam superprovisionar continuamente o sistema.

Em estudo que analisou 24 horas de tráfego recebido pelo servidor raiz F no dia 04 de outubro de 2002, identificou-se uma quantidade exorbitante de consultas idênticas (25,4%) e repetidas (44,9%) que foram recebidas por uma de suas 4 instâncias. Apontou-se que consultas ilegítimas (endereço de origem em faixa de rede de uso privado, consultas para um nome que já é um IP, consultas repetidas, idênticas e para as quais o resolvidor deveria ter feito *cache* ou mesmo com caracteres inválidos) totalizam 97,85% de todas as consultas recebidas [18].

Em trabalho no qual se analisou requisições recebidas pelo *Root Server D*, antes, durante e por 7 meses após a alteração de seu endereço IP (realizada em 03 de janeiro de 2003), constatou-se que uma grande quantidade de consultas recebidas, tanto antes da mudança de endereço quanto depois, possuem problema. Embora não tenham sido apresentados percentuais em relação ao total, é possível inferir que, da mesma forma que em trabalhos anteriormente citados, os principais motivos por trás do defeito dessas consultas é a solicitação de TLDs inválidos ou requisições malformadas [51].

Quanto à utilização de GMM para clusterização, cabe citar o trabalho realizado por Kai *et al.* cujo intento era o de conceber e construir um modelo de detecção para identificar anomalias de tráfego de rede que envolvem *Big Data* em substituição a utilização de IDS. Após a utilização de K-means e GMM, foi possível perceber que, enquanto o K-means falhou em detectar até mesmo uma única anomalia no conjunto de dados simulado, provando não ser complexo o suficiente para lidar com diversidades do tráfego DNS, o GMM foi capaz de detectar as anomalias com uma taxa de acerto de 95% usando-se dois *clusters*. Após a atualização do número de componentes para 5, o modelo GMM final alcançou uma taxa de detecção de 100% nas anomalias de exfiltração de dados [52].

Outro estudo [42] também adotou a utilização de GMM para detectar tráfego DNS anormal tendo constatado que este modelo é adequado para cenários *Big Data* e fornece resultados de maneira eficiente desde que os *outliers* sejam identificados e removidos com sucesso nos dados de treinamento a fim de não aumentarem a variabilidade dos dados, o que permitiria que *outliers* semelhantes nos dados de teste passassem sem serem detectados.

Embora pesquisas anteriores já tenham investigado servidores DNS, legitimidade de tráfego recebido, configurações de TTL e alguns padrões de uso de DNS de forma detalhada, implementações e comportamentos de resolvidores recursivos não são bem conhecidos. Também, não há metodologia que ajude a compreender e identificar quais resolvidores são agressivos. Assim, ciente da falta de entendimento e compreensão de quem são os resolvidores com comportamento agressivo, a pesquisa descrita neste trabalho busca quantificar e classificar os comportamentos recursivos dos resolvidores, especi-

almente daqueles que enviam consultas em maior quantidade que o tolerável segundo TTL sugeridos.

3. PROCEDIMENTOS METODOLÓGICOS

Realizada a revisão de literatura e estudo acerca dos componentes envolvidos no sistema DNS, este capítulo destina-se a apresentar e detalhar os recursos utilizados e procedimentos adotados que permitiram o desenvolvimento de uma solução (metodologia) capaz de identificar e classificar resolvedores com comportamento abusivo, objetivo geral dessa pesquisa.

3.1 DATASETS DITL

Conforme referido no capítulo 2, para o desenvolvimento do trabalho foram utilizados *datasets* do projeto *Day In The Life of the Internet* (DITL), disponibilizados pela DNS-OARC.

Esses *datasets* DITL contêm as consultas realizadas geralmente por resolvedores recursivos e recebidas pelos servidores raiz, em um período de cerca de 48 horas contínuas, a cada ano. A coleta normalmente se inicia por volta das 11 horas de um dia 'd' estendendo-se até o meio-dia e 55 minutos do dia 'd+2'. Assim, embora as coletas ocorram em três dias por ano, isso acontece porque o projeto visa capturar um dia da internet relativo a cada fuso horário existente no mundo. Entretanto, será utilizado, por padrão, o horário GMT como base para as 24 horas processadas. Desta forma, a fim de se obter estimativas iniciais em relação aos dados disponibilizados pelo DNS-OARC, para cada conjunto de dados DITL, foram extraídos todos os pcaps do dia completo para aquela coleta, extraíndo-se a informação para qual servidor raiz foi feita a requisição, a data e hora da consulta, o tipo de consulta (recursos do tipo A, AAAA, NS ou DS), o endereço requisitado considerando-se os domínios de nível superior (TLD) .com, .cn e .nl e o IP que originou a consulta.

A Tabela 4 apresenta estatísticas resumidas dos *datasets* DITL utilizados. Optou-se pela utilização dos dados dos últimos 5 anos. Além de representar um bom espaço amostral, esse período apresenta uma estabilização quanto aos *datasets* participantes do projeto. Esta similitude entre os *datasets* permite com que se alcance um resultado mais fidedigno, sem muita distorção proveniente da ausência de participação de *root servers* em algum ano.

Conforme é possível extrair da tabela apresentada, independente do ano, não existem dados relacionados ao *root server* letra 'G'. Outro fator a ser analisado é o crescente tamanho do *dataset* aos longo dos anos, o que indica não apenas a participação de instâncias dos servidores raiz mas, principalmente, o aumento no número de requisições recebidas por todos esses *root servers*.

Tabela 4: Informações gerais dos *datasets* DITL utilizados. Fonte: Autora.

Período da coleta	Servidores raízes ausentes	Tamanho do <i>dataset</i>
05-07/04/2016	D-root e G-root	4T
11-13/04/2017	G-root	5.9T
10-12/04/2018	G-root	5.9T
08-10/04/2019	B-root e G-root	7.5T
05-07/05/2020	G-root	11T

Através de *script* em *bash* foi realizada extração, coletando-se em relação às requisições recebidas no dia intermediário do período de coleta, o *root* que recebeu a solicitação, o *timestamp* da requisição, o tipo de recurso requisitado, o IP do resolvidor que fez a requisição e o domínio consultado. As informações foram agregadas em linhas, cada qual representando informações acerca de uma requisição específica, e consolidadas em arquivos *csv* (*comma-separated values*) de acordo com a letra do *root server* que recebeu as solicitações. O formato de linha pode ser demonstrado através do exemplo a seguir:

```
a,1554768117.630,43,211.63.64.11,0051d0b5.cdn.uc1oud.cn
```

Em virtude da enorme quantidade de dados presentes em cada DITL, procurou-se avaliar os comportamentos comuns dos resolvidores escolhendo-se solicitações de tradução para os TLDs *.nl*, *.cn* e *.com*. A escolha destes domínios de nível superior se deu devido ao fato do *.com* ser o maior domínio gTLD, o *.cn* ser um dos maiores ccTLDs do mundo e do *.nl* poder ser associado aos dados obtidos pelo SIDN ou Fundação de registro de domínio da Holanda (*Stichting Internet Domeinregistratie Nederland*) em uma futura comparação dos resultados encontrados por este estudo com os dados obtidos por esse instituto [53, 54].

Já na extração, analisou-se cada consulta a fim de excluir aquelas que não contivessem a quantidade de campos desejados, aqueles em que o campo reservado para o IP de origem não correspondesse à forma desejada e aquelas com domínio inválido e não correspondente a algum dos TLDs adotados na pesquisa. Posteriormente, realizou-se a leitura e análise dos arquivos a fim de se gerar estatísticas de quantidade de consultas por servidor raiz, funções de distribuição acumulada das requisições, identificação dos IPs que mais realizaram consultas (tratados neste trabalho como resolvidores) e distribuição das consultas realizadas por esses resolvidores ao longo do dia de coleta. A Tabela 10 apresenta estatísticas resumidas após a análise preliminar dos dados dos intervalos de 24 horas mais completos dos últimos anos de rastreamentos dos servidores DNS raiz disponíveis.

Tabela 5: Informações da coleta de dados nos servidores de nomes DNS. Fonte: Autora.

Data utilizada	Quant. de dados	Quant. de consultas	Quant. de resolvedores únicos
05/04/2016	79GB	7.163.357.401	7.396.026
12/04/2017	88GB	7.157.607.184	8.527.792
11/04/2018	91GB	7.904.917.521	9.603.349
09/04/2019	94GB	7.746.604.264	8.859.099
06/05/2020	97GB	8.702.203.719	12.332.252

3.1.1 Limitações na captura e processamento dos dados

Cabe esclarecer que, como de forma geral os dados não podem ser retirados do servidor da OARC, e como não existe um banco de dados à disposição dos pesquisadores, para toda a primeira parte do trabalho (extração e tratamento) utilizou-se, quase que exclusivamente, ferramentas como TShark⁴ e *scripts* em *bash* (*Bourne-Again Shell*) que, por lerem uma imensa quantia de dados a cada vez (e muitas vezes os mesmos dados embora agrupados diferentemente de acordo com o que se espera extrair) foram de lenta execução, estando condicionados a um servidor cujo processamento e armazenamento é compartilhado com outros usuários.

Atualmente, o servidor conta com 1.8 *terabytes* de disco disponibilizado aos pesquisadores. Embora estes 1.8 *terabytes* possam parecer uma quantidade razoável, é preciso lembrar que o espaço é compartilhado dentre diversos pesquisadores, cada qual minerando *datasets* de muitos *terabytes* conforme já referido. Com isso, em várias oportunidades foi necessário descartar resultados ou dados coletados haja vista a ausência de espaço em disco. Também, foi necessário que se trabalhasse sempre com arquivos zipados e que, na medida em que os dados fossem transformados, os arquivos de origem fossem apagados a fim de que se liberasse espaço para os processamentos.

Tendo em vista a impossibilidade de guarda dos arquivos em que inicialmente foram consolidadas as informações como forma de *backup*, e tendo em vista as instabilidades diversas vezes observadas no servidor, houve em diversos momentos a ocorrência de erros e falhas em *scripts* o que acabou ocasionando a perda de informações e um lento retrabalho.

Outro impacto a ser citado é a manutenção periódica do servidor a cada duas semanas e a reinstalação do mesmo que acabou por alterar a versão de diversas ferramentas base do *bash*, como por exemplo o AWK⁵ que ficou sem diversas opções de comando e

⁴TShark é um analisador de protocolo de rede que permite capturar dados de pacote de uma rede ativa ou mesmo ler pacotes de um arquivo de captura salvo anteriormente. Com ele é possível imprimir em uma forma decodificada desses pacotes na saída padrão ou gravar os pacotes em um arquivo. O seu formato de captura nativa é o pcapng [55].

⁵AWK é uma linguagem de programação criada em 1977 que permite processar textos e manipular arquivos. Por ser interpretada, não precisa de compilação.

com limite em *big integers*, o que impactou no funcionamento de diversos *scripts* que eram adotados até então.

Dadas as frequentes instabilidades do servidor e o constante prejuízo de consistência nos dados coletados em virtude dessas indisponibilidades e alterações ou limitações das ferramentas ali oferecidas, ou até mesmo da alteração de estratégia de dados utilizados para o desenvolvimento da pesquisa, ao longo de quase dois anos, por diversas vezes realizou-se a extração, mineração, tratamento e transformação dos dados do zero, ou seja, desde o princípio. Com isso, ao longo do tempo, inúmeros *scripts* até então utilizados foram descontinuados ou mesmo reformulados. Enquanto que a grande maioria dos *scripts* inicialmente utilizados eram *shell script* com uso de AWK, posteriormente, passou-se a utilizar *scripts* em *python*. Ainda assim, o constante retrabalho e adequações permitiram que a extração e o tratamento dos dados fosse aperfeiçoado e aprimorado.

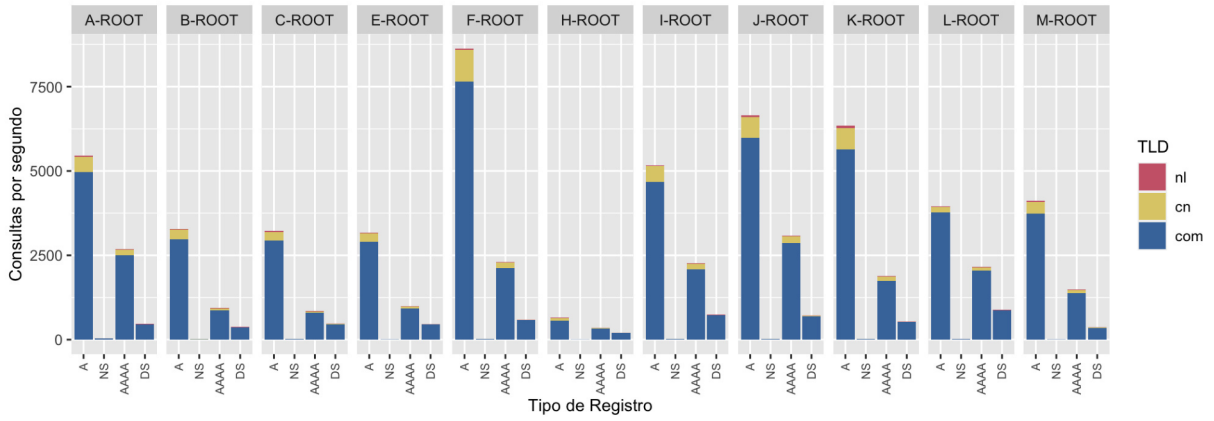
Após um ano e alguns meses de pesquisa foi permitida a extração de arquivos contendo dados consolidados de consultas realizadas por resolvedores devidamente anonimizados do servidor. Isso permitiu a utilização de banco de dados e a posterior definição dos atributos a serem utilizados no processo de classificação dos resolvedores.

3.2 DISTRIBUIÇÃO INICIAL DOS DADOS

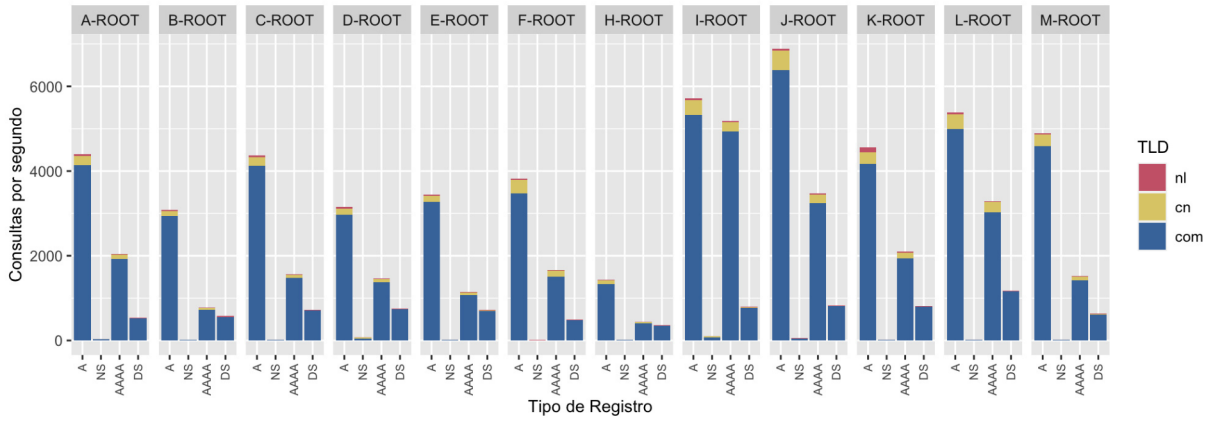
Inicialmente, procurou-se descobrir qual a distribuição dos dados no que se refere às letras dos servidores raízes, tipo de recurso solicitado e TLD requerido, ao longo dos anos dos *datasets* DITL escolhidos.

No que tange às letras dos *roots server* que receberam as solicitações, verificou-se que há uma grande diferença de um ano para outro. Isso se dá não apenas porque ao longo dos anos alguns servidores raiz ganham mais réplicas em diferentes localidades ao redor do mundo, mas também porque nem todas essas réplicas participam das coletas a cada ano ou por todo o período de coleta do DITL.

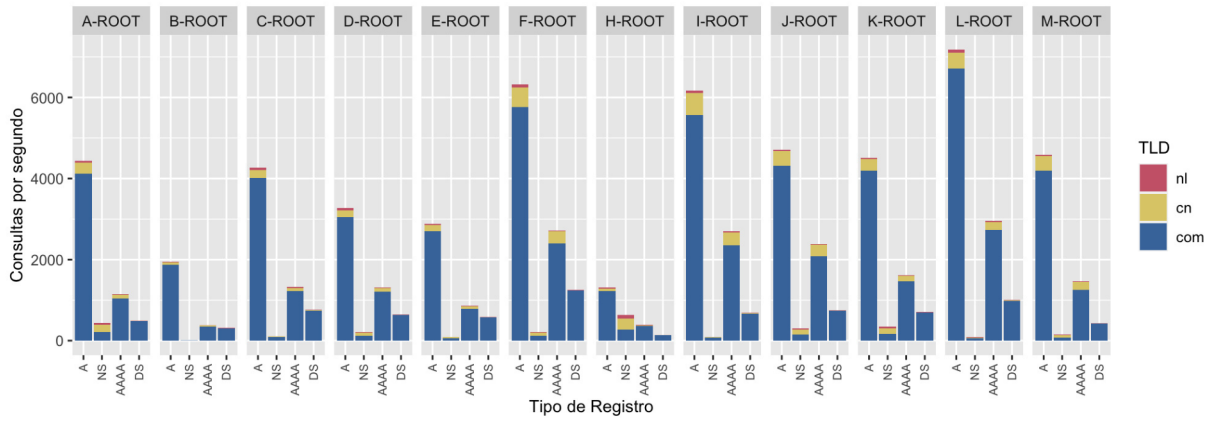
Em relação aos dados de 2016 a 2020 utilizados no presente trabalho, as distribuições de acordo com as letras de *root* que receberam informações, correspondem a:



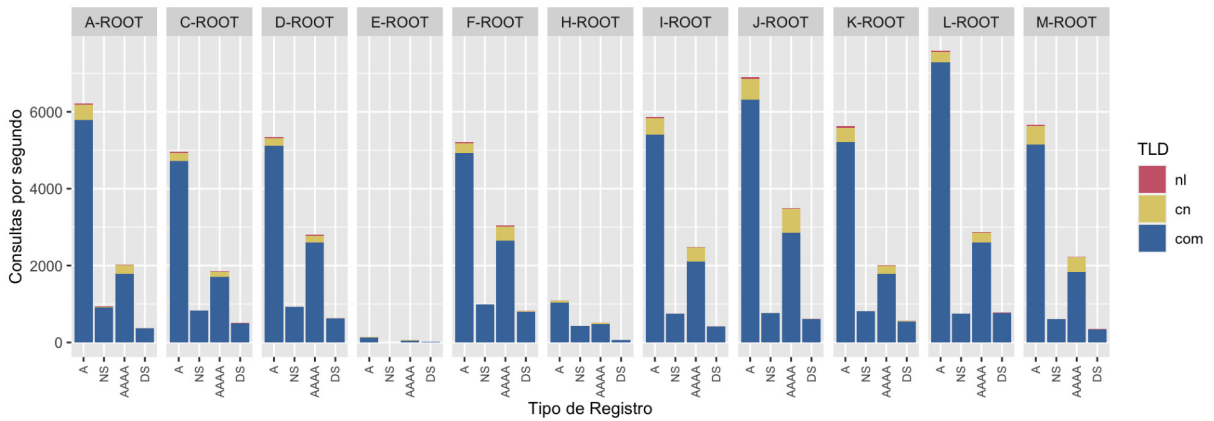
(a)



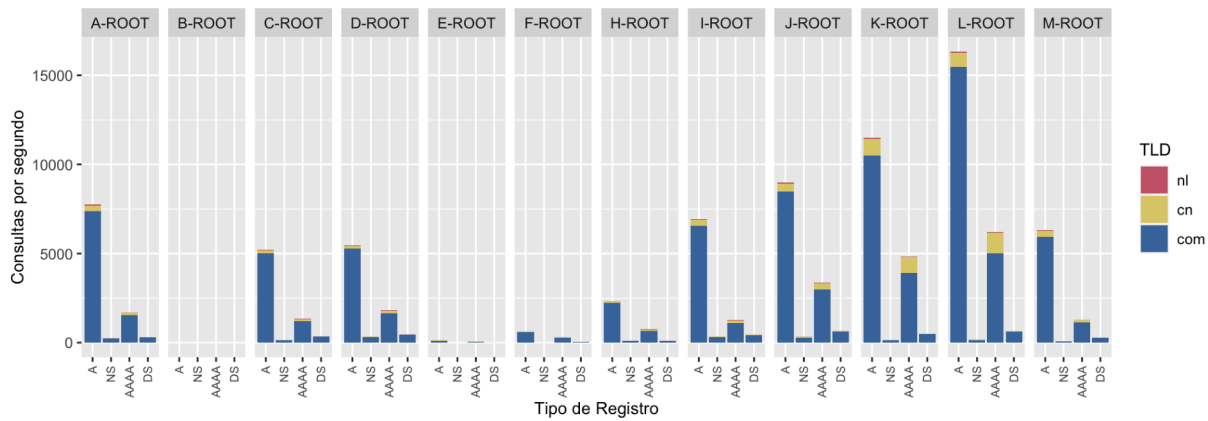
(b)



(c)



(d)



(e)

(a) DITL 2016, (b) DITL 2017, (c) DITL 2018, (d) DITL 2019, (e) DITL 2020

Figura 2: Consultas recebidas pelos *root servers* em diferentes anos divididas por TLD e tipo de recurso requisitado. Fonte: Autora.

Os gráficos apresentados trazem todas as requisições recebidas pelos *root servers* em cada ano do projeto DITL, resultantes da extração realizada que teve seus critérios explicitados anteriormente. As requisições estão primeiramente divididas pela letra do servidor que as recebeu e também pelo tipo de recurso requisitado (A, NS, AAAA, DS). Quanto ao tipo de recurso solicitado, estão divididas proporcionalmente de acordo com o TLD requisitado (.cn, .nl, .com).

É possível notar que para o ano de 2020 houve um aumento significativo do número de requisições recebidas principalmente pelos *root servers* letras 'L', 'K' e 'J'. Já a letra 'B' não está presente no DITL de 2019 e pouco aparente no DITL de 2020 (Não foram disponibilizados dados suficientes relacionados ao B-root para este ano). Também, pelas imagens indicadas é possível notar que a esmagadora maioria das requisições são para o TLD .com (cerca de 92% independentemente do ano do DITL). Quanto ao tipo de recurso mais solicitado, corresponde ao A (66% em 2016, 60% em 2017, 62% em 2018, 60% em 2019 e 71% em 2020), seguido do recurso AAAA (25% em 2016, 29% em 2017, 24% em 2018, 25% em 2019 e 23% em 2020). O recurso DS foi o terceiro mais requisitado em todos os anos apurados com exceção de 2019, onde o terceiro mais requerido foi o NS. Esses dados podem ser melhor observados através das Tabelas 6, 7 e 8.

É importante ressaltar que, nas Tabelas 6, 7 e 8, enquanto a soma total das requisições corresponde efetivamente ao total de requisições para cada DITL, o somatório dos resolvedores não correspondem ao total de resolvedores para cada DITL. Isso porque, uma consulta será sempre apenas para uma ou outra letra de servidor raiz, ou solicitando um ou outro recurso, contudo, um mesmo resolvedor poderá realizar diversas consultas, cada uma para uma letra de *root server* diferente ou solicitando endereços com diferentes TLDs. Para verificação de quantidade de diferentes resolvedores que tenham realizado consulta para cada ano DITL, deverá ser consultada a Tabela 10.

Tabela 6: Estatísticas das requisições recebidas agrupadas por tipo de recurso requisitado. Fonte: Autora.

RR	2016			2017			2018			2019			2020		
	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.
A	6.6M	4.7B	66,1	7.5M	4.2B	60	8.6M	4.9B	62,5	7.8M	4.6B	60,1	11M	6.2B	71,5
AAAA	2.5M	1.8B	25,5	2.8M	2.1B	29,5	3.2M	1.8B	24	3.1M	1.9B	25,5	3M	1.9B	22,9
NS	113K	16M	0,2	362K	33M	0,5	388K	260M	3,3	461K	674M	8,7	246K	165M	1,9
DS	583K	582M	8,1	652K	719M	10	738K	806M	10,2	704K	442M	5,7	641K	323M	3,7
Total	9.8M	7.1B	100	11M	7.1B	100	13M	7.9B	100	12M	7.7B	100	15M	8.7B	100

Tabela 7: Estatísticas das requisições recebidas agrupadas por TLD requisitado. Fonte: Autora.

TLD	2016			2017			2018			2019			2020		
	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.
.com	6.9M	6.5B	91,7	8M	6.7B	93,7	9M	7.2B	92	8M	7.1B	92,83	11.8M	8B	92,4
.cn	2M	540M	7,5	2M	378M	5,3	2M	541M	6,9	1M	512M	6,62	2.1M	604M	6,9
.nl	1.3M	55M	0,8	1M	72M	1,0	1M	89M	1,1	1M	42M	0,55	1.1M	55M	0,6
Total	10M	7.1B	100	11M	7.1B	100	12M	7.9B	100	11M	7.7B	100	15M	8.7B	100

Tabela 8: Estatísticas das requisições recebidas agrupadas por letra de servidor raiz que recebeu a requisição. Fonte: Autora.

Root Server	2016			2017			2018			2019			2020		
	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.	qde. resol.	qde. req.	% req.
A	2.1M	800M	11,2	2.1M	595M	8,3	2.4M	605M	7,7	2.4M	807M	10,4	2.9M	867M	10
B	1.6M	426M	6	1.6M	381M	5,3	1.3M	265M	3,4	x	x	x	3	271	0
C	1.7M	419M	5,9	1.9M	566M	7,9	2.1M	645M	8,2	1.9M	694M	9	3.2M	612M	7
D	x	x	x	1.3M	461M	6,5	1.6M	533M	6,8	1.5M	827M	10,7	2.2M	702M	8,1
E	1.5M	403M	5,6	1.8M	453M	6,3	1.3M	423M	5,4	82K	19M	0,3	173K	16M	0,2
F	1.3M	1B	14,6	814K	486M	6,8	1.3M	1.1B	14,3	749K	870M	11,2	98K	85M	1
H	358K	128M	1,8	709K	190M	2,7	977K	246M	3,1	813K	177M	2,3	1M	284M	3,3
I	1.5M	792M	11,1	1.8M	1B	14,1	1.9M	925M	11,7	1.5M	812M	10,5	2.3M	782M	9
J	1.6M	1.1B	15,3	1.8M	939M	13,1	2M	767M	9,7	1.9M	1B	12,9	2.5M	1.1B	13,3
K	1.6M	818M	11,4	1.8M	636M	8,9	2M	702M	8,9	1.9M	756M	9,8	3.3M	1.4B	17
L	1.3M	665M	9,3	1.6M	841M	11,8	1.7M	1B	13,2	1.7M	1B	13,4	2.4M	2B	23,2
M	1.6M	566M	7,9	1.8M	593M	8,3	2M	617M	7,8	1.9M	743M	9,6	3.3M	693M	8,0
Total	16M	7.1B	100	19M	7.1B	100	21M	7.9B	100	16M	7.7B	100	24M	8.7B	100

3.3 DISTRIBUIÇÃO DAS CONSULTAS POR RESOLVEDORES

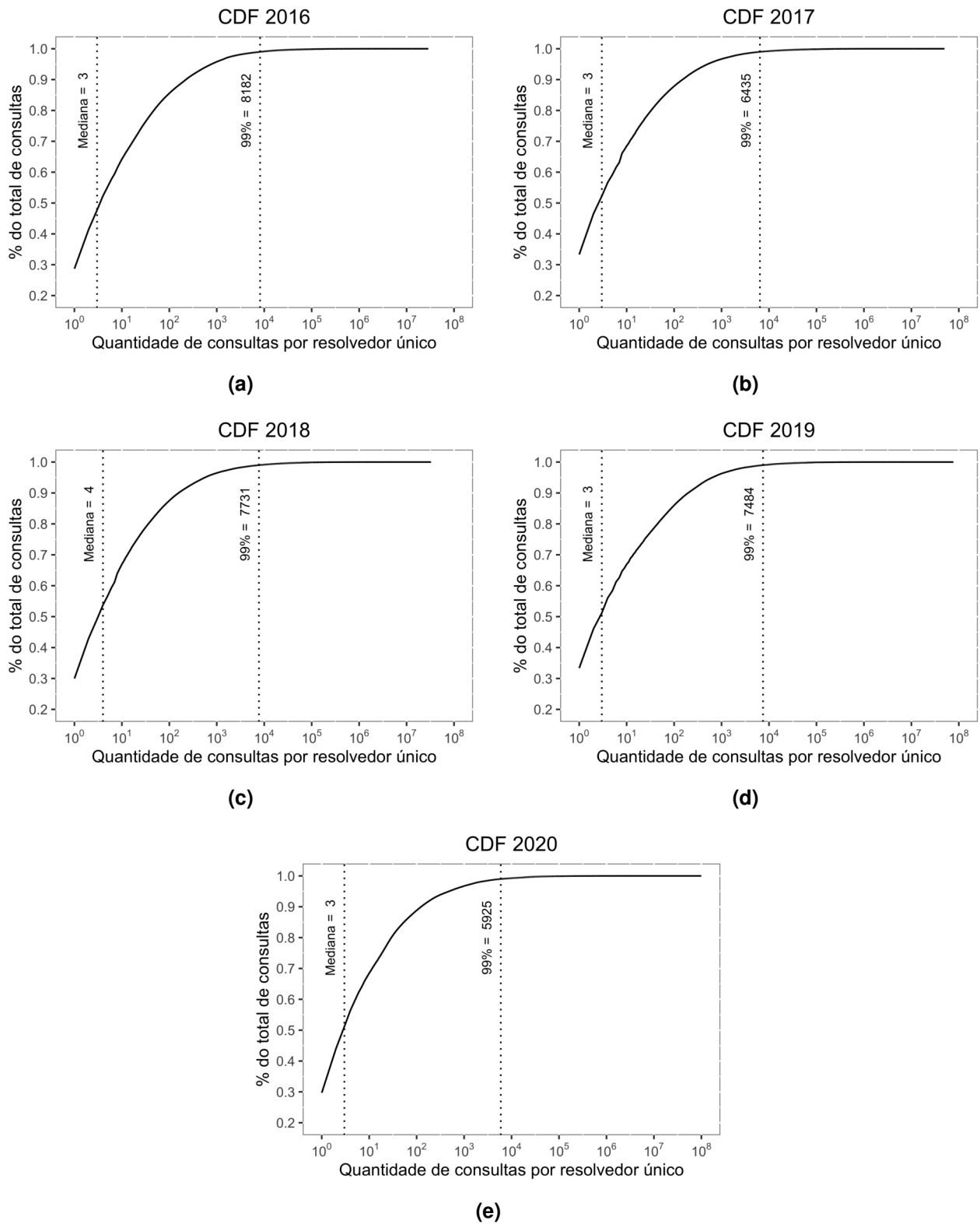
Posteriormente à verificação de distribuição das consultas segundo os critérios de letra de servidor raiz, TLD e RR, as consultas foram reagrupadas em arquivos de acordo com o resolvedor IP que as fez, momento em que se observou que uma quantidade considerável de resolvedores enviou mais solicitações do que o esperado, e muito mais solicitações do que o tolerável, considerando-se o valor típico para os TTLs daqueles registros de recursos escolhidos.

Conforme é possível notar da Figura 3, durante as 24 horas utilizadas de captura dos DITL de 2016 até 2020, cerca de 33% dos resolvedores enviaram apenas uma consulta diária e cerca de 50% deles enviaram três ou quatro consultas no dia, considerando todos os servidores raiz da estrutura do DNS. Também, os 99% resolvedores que menos enviaram requisições para toda a estrutura raiz do DNS enviaram uma quantidade de menos de 6 mil requisições, cada um, para o ano de 2020, de cerca de 6 mil requisições, cada um, para o ano de 2017, e, em torno de, 7 mil requisições, cada um, para os dias do DITL 2018 e 2019, e, finalmente, mais de 8 mil requisições, cada um, para o DITL de 2016.

Além disso, quando selecionados apenas os 1% resolvedores que mais consultaram, descobriu-se que eles são responsáveis pelo envio de até 87% de todo o tráfego, nos anos de 2016, 2018 e 2019, de até 88% de todo o tráfego para o ano de 2020 e de até 89% de todo o tráfego para o ano de 2017. Esta quantidade é alarmante tendo-se em vista que, para os dados coletados, o 1% de resolvedores recursivos com este comportamento aparentemente abusivo corresponde a cerca de 7,3 milhões de recursivos em 2016, 8,5 milhões em 2017, 9,6 milhões em 2018, 8,8 milhões em 2019 e mais de 12 milhões de recursivos em 2020, o que indica um número muito grande de resolvedores recursivos abusivos na Internet, fazendo mau uso da infraestrutura DNS.

Considerando-se que apenas se está avaliando apenas os 1% “piores” resolvedores no que se refere à quantidade de consultas realizadas e que, ainda assim, apenas esses 1% são responsáveis por cerca de 90% do tráfego registrado para os dias analisados, é possível inferir o quanto de consultas ilegítimas chegam aos *root servers*. Nesta enorme quantidade de consultas enviadas por esses recursivos abusivos fica evidente a existência de tráfego DNS potencialmente inútil, desperdiçando recursos, por vezes críticos, da infraestrutura raiz do DNS.

Embora fosse esperado que o número de requisições enviadas por um único resolvedor por dia fosse muito mais baixo do que valores próximos a 7 ou 8 mil requisições, já que o TTL previsto para os recursos A, NS, AAAA é de 172800 segundos (equivalente a 2 dias) e o do recurso DS é de 86400 (equivalente a 1 dia) [56], na prática, verificou-se que alguns resolvedores enviaram ainda mais requisições do que os números já entendidos como elevados.



(a) DITL 2016, (b) DITL 2017, (c) DITL 2018, (d) DITL 2019, (e) DITL 2020

Figura 3: Requisições acumuladas de acordo com resolvidores únicos ao longo dos diferentes DITL. Fonte: Autora.

Há casos extremos, em que um único resolvidor enviou uma média de 891 consultas por segundo sozinho para toda a estrutura raiz de DNS em 2019 e 1853 consultas em 2020 (Tabela 9).

Na Tabela 9 são apresentadas as quantidades de requisições realizadas pelos dez resolvedores que mais realizaram consultas para os DITL de 2016 a 2020. Conforme é possível constatar, qualquer um dos resolvedores presentes na tabela realiza, por segundo e em média, uma quantidade maior de requisições que seria esperada receber de um único resolvedor durante todo o dia. A ausência de utilização apropriada de *cache*, má configuração dos resolvedores ou mesmo utilização de funções *lambda*⁶ para criar resolvedores que em pouco tempo serão destruídos juntamente com seus *caches*, entre outros motivos, implicam na existência de resolvedores que realizam, sozinhos, uma enorme quantidade de requisições para a estrutura DNS.

Tabela 9: TOP 10 resolvedores IP referente aos DITL de 2016, 2017, 2018, 2019 e 2020. Fonte: Autora.

Pos	2016			2017			2018			2019			2020		
	% do total	qde. req.	qde p/ seg.	% do total	qde. req.	qde p/ seg.	% do total	qde. req.	qde p/ seg.	% do total	qde. req.	qde p/ seg.	% do total	qde. req.	qde p/ seg.
1	0,4	28M	332	0,7	49M	578	0,4	32M	379	1,0	76M	891	1,8	160M	1853
2	0,4	26M	310	0,4	28M	329	0,4	28M	325	0,9	71M	822	1,7	144M	1673
3	0,4	25M	291	0,4	27M	322	0,3	24M	288	0,9	70M	812	1,1	99M	1152
4	0,3	24M	280	0,3	20M	242	0,3	21M	253	0,7	56M	649	0,9	74M	859
5	0,3	22M	261	0,2	15M	183	0,3	21M	244	0,7	54M	628	0,5	42M	493
6	0,3	20M	242	0,2	14M	164	0,3	20M	242	0,6	46M	533	0,3	28M	325
7	0,3	20M	241	0,2	13M	161	0,3	20M	235	0,4	31M	358	0,3	26M	306
8	0,3	19M	224	0,2	13M	161	0,2	16M	194	0,3	23M	275	0,3	25M	299
9	0,3	18M	210	0,2	13M	161	0,2	16M	186	0,3	23M	274	0,3	23M	276
10	0,2	17M	204	3,0	13M	160	0,2	14M	167	0,3	22M	265	0,3	22M	257
Total	3,1	224M	2596	212M	3,0	2461	2,7	217M	2513	6,1	475M	5506	7,4	305M	7492

Conforme se pode deduzir das tabelas apresentadas até o momento, embora a quantidade média de consultas por resolvedor tenha diminuído (já que a quantidade de consultas aumentou menos ao longo dos DITL em proporção ao aumento de resolvedores ao longo desses mesmos conjuntos de dados), os resolvedores que mais fizeram consulta (Top 10) aumentaram consideravelmente a quantidade de requisições realizadas por segundo (de 332 em 2016 para 1853 em 2020). Também é possível perceber que eles passaram a ser responsáveis por uma maior fatia do total das requisições recebidas pelos *root servers* naquele ano (de 3,1% em 2016 para 7,4% em 2020).

Cabe comentar que o ano de 2020 foi marcado pela continuidade de pandemia de corona vírus instaurada no final de 2019 (COVID-19), a qual trouxe diversos impactos sociais, econômicos, culturais e políticos para todos os países do globo e que, inevitavelmente, refletiram em um aumento no volume de tráfego na Internet.

⁶Funções como serviço são utilizadas como um recurso de computação em nuvem que permitem a execução de código sem que seja necessário provisionar ou gerenciar servidores. Com isso, um serviço com resolução de endereços pode ser facilmente instanciado e, em pouco tempo interrompido.

É provável que o aumento da quantidade de resolvedores e do número de consultas realizadas pelos resolvedores que mais tenham feito requisições seja consequência das necessidades advindas com a pandemia e das mudanças culturais acarretadas por ela, muitas destas persistentes até a presente data.

3.4 TOP 30 RESOLVEDORES COM MAIOR QUANTIDADE DE REQUISIÇÕES

Em estudo inicial, foram analisados os 30 resolvedores com maior quantidade de consultas realizadas, para cada DITL, procurando-se encontrar um padrão no envio de suas consultas, assim como procurou-se entender a distribuição dessas mesmas consultas ao longo do dia. Dito isso, as figuras 4 a 8 apresentam a distribuição dos IPs que mais realizaram consultas ao longo do dia de coleta utilizado, para os anos de 2016 a 2020.

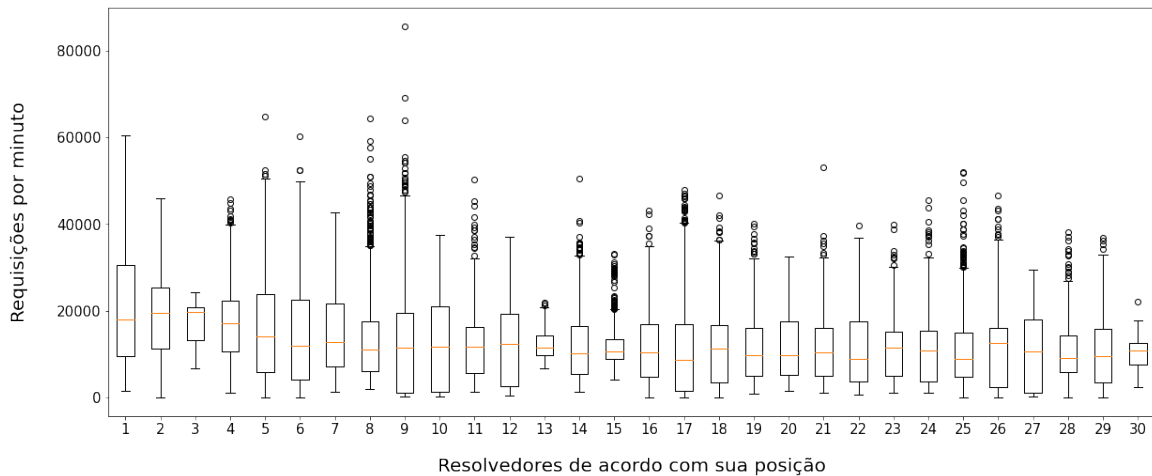


Figura 4: Distribuição quantitativa das consultas dos 30 resolvedores com maior número de requisições para o ano de 2016. Fonte: Autora.

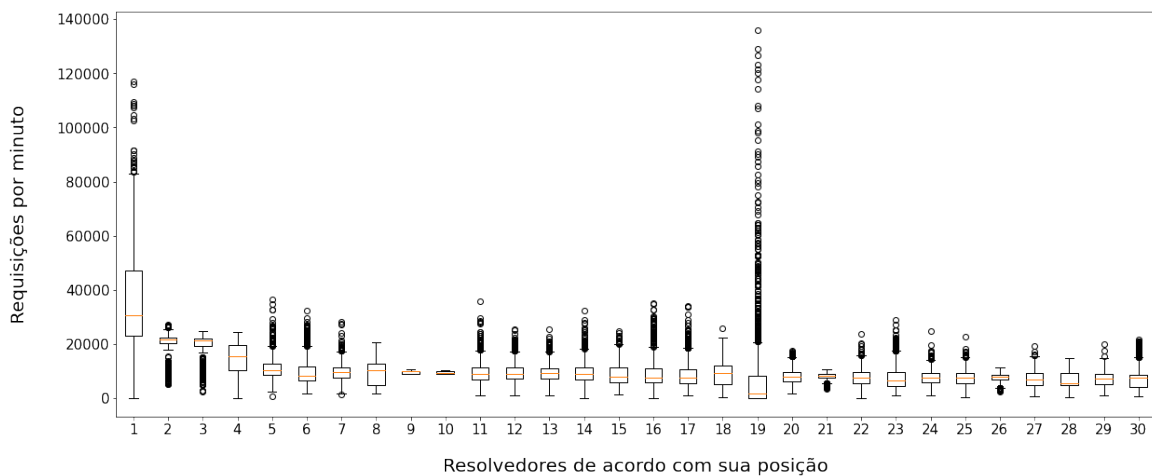


Figura 5: Distribuição quantitativa das consultas dos 30 resolvedores com maior número de requisições para o ano de 2017. Fonte: Autora.

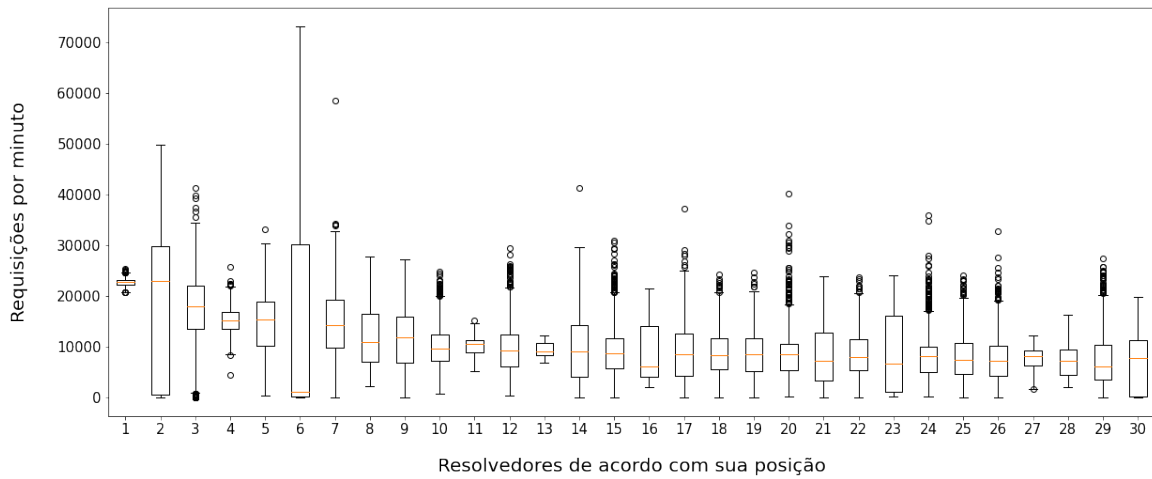


Figura 6: Distribuição quantitativa das consultas dos 30 resolvedores com maior número de requisições para o ano de 2018. Fonte: Autora.

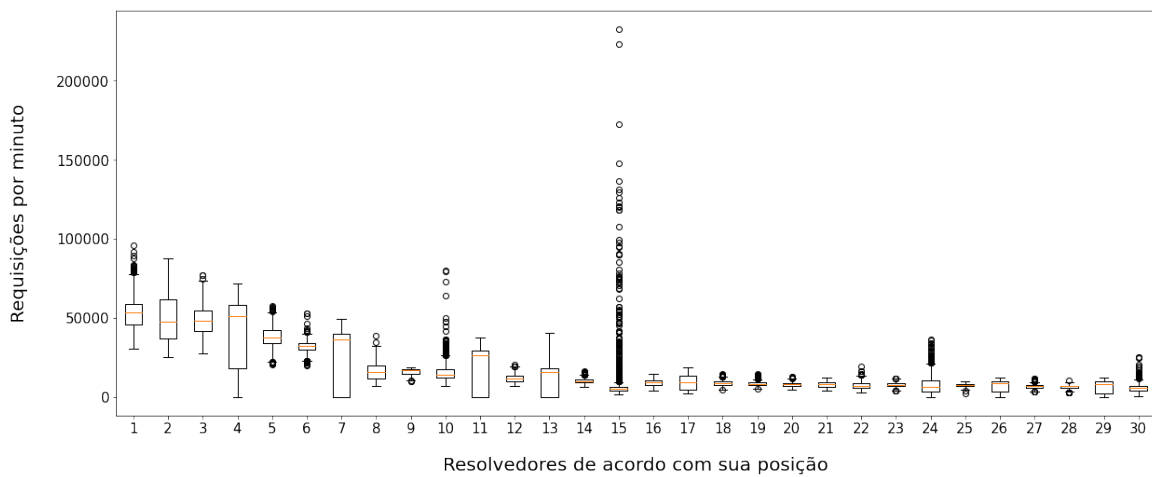


Figura 7: Distribuição quantitativa das consultas dos 30 resolvedores com maior número de requisições para o ano de 2019. Fonte: Autora.

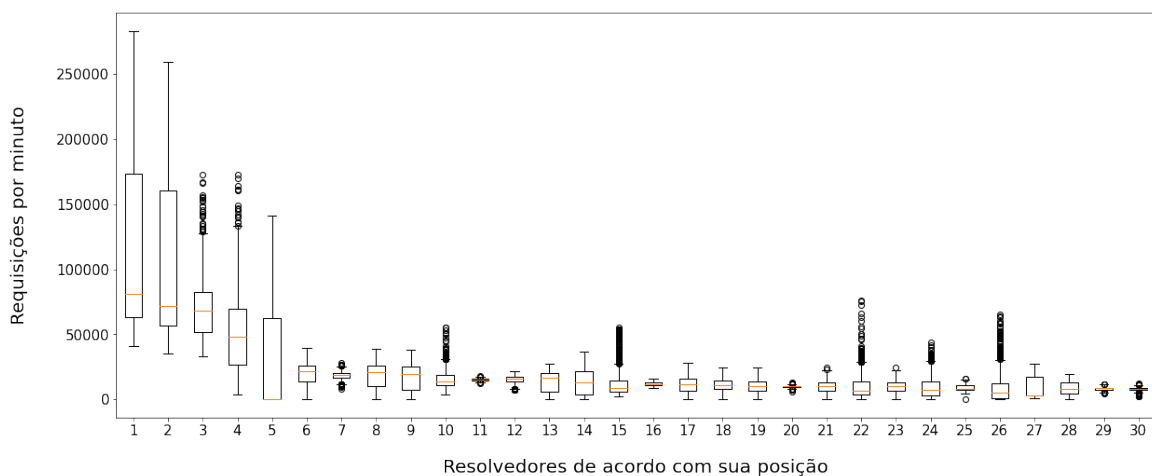
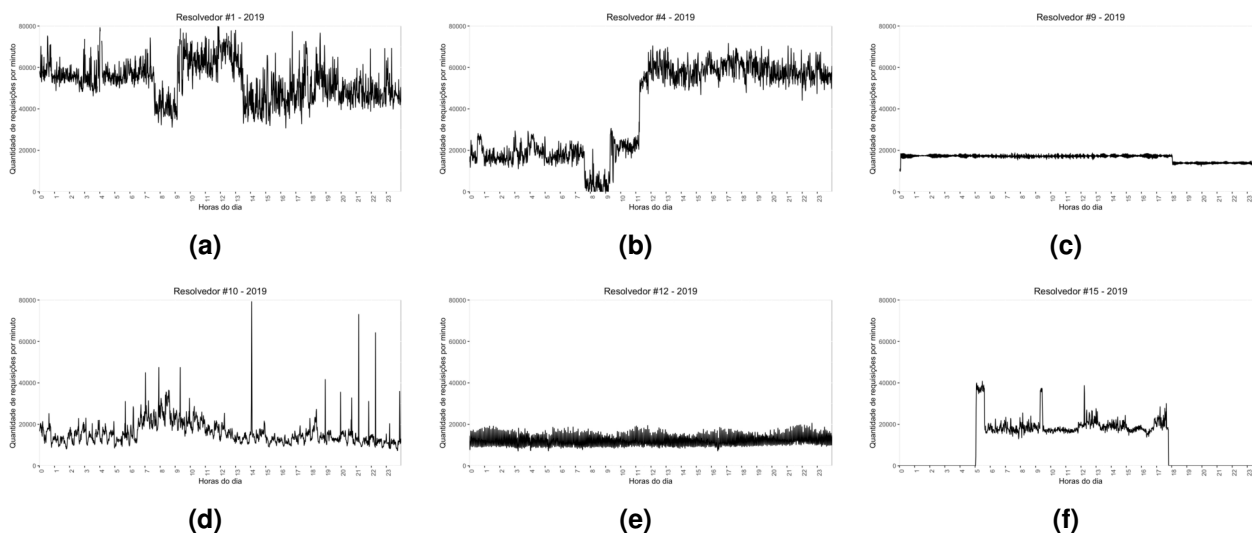


Figura 8: Distribuição quantitativa das consultas dos 30 resolvedores com maior número de requisições para o ano de 2020. Fonte: Autora.

É possível identificar algumas semelhanças dentre alguns resolvedores, contudo, é notória a enorme diversidade de comportamento entre eles, o que não permite, prontamente e precipitadamente, identificar grupos. A fim de demonstrar a diversidade de comportamento dos resolvedores, apresenta-se a distribuição de alguns dos TOP 30 resolvedores com mais requisições do DITL de 2019 e presentes na Figura 9.



(a) Resolver #1 **(b)** Resolver #4 **(c)** Resolver #9 **(d)** Resolver #10 **(e)** Resolver #12 **(f)** Resolver #15

Figura 9: Estrutura em níveis do sistema DNS. Fonte: Autora.

Conforme pode ser verificado, alguns resolvedores possuem uma média uniforme nas requisições feitas ao longo do dia analisado enquanto outros, por sua vez, mantiveram uma média muito mais alta em determinados momentos e em outra parte do dia realizaram um número consideravelmente menor de requisições. Existem ainda resolvedores que possuem diversos picos, podendo estes representarem muitas vezes o valor de sua média de consultas ou mesmo poucos desvios-padrão acima.

A diversidade observada na distribuição das consultas dos resolvedores ao longo do dia dos DITL apurados e a variabilidade na quantidade de requisições realizadas favoreceu o entendimento de que a forma mais eficaz de os identificar e classificar poderia ser através de aprendizagem de máquina.

3.5 PREPARAÇÃO DOS DADOS PARA CLASSIFICAÇÃO

Tendo em vista a natureza dos *datasets*, dos dados coletados e, principalmente a diferença no envio de consultas pelos resolvedores ao longo dos dias de DITL analisados, optou-se pela utilização de aprendizado de máquina a fim de que se pudesse classificar os resolvedores e identificar aqueles cujo comportamento impacta sobremaneira os *root servers*. Haja vista a inexistência de rótulo nos dados, requisito necessário para a utilização de

O primeiro campo da linha corresponde ao endereço IP do resolvidor anonimizado pela aplicação de algoritmo de *hash* SHA256⁷. O segundo campo é a quantidade total de requisições feitas por aquele dado resolvidor. O terceiro campo é a lista das requisições agrupadas a cada um minuto. O quarto elemento é quantas requisições há por letra de *root server* e, finalmente, o quinto e o sexto campos correspondem, respectivamente, à quantidade de consultas divididas por tipo de recurso e TLD requisitado. Este arquivo csv possui tantas linhas quantos resolvidores existentes para cada ano de DITL e é zipado a fim de ser retirado do servidor através de comando *scp*⁸.

Uma vez retirado o arquivo com os dados consolidado do servidor ele é descompactado e tratado a fim de que suas linhas sejam lidas e convertidas em documentos em banco de dados MongoDB⁹. Através de um *script* também em *python*, são geradas algumas estimativas e atributos a serem utilizados no processo de clusterização.

Inicialmente, é calculada a média das requisições agrupadas por minuto, o desvio padrão dessas mesmas requisições agrupadas por minuto, a quantidade de picos na distribuição das consultas e a quantidade de minutos ativos (quantidades de valores na lista em que o valor seja diferente de zero). Cabe dizer que no cálculo de média e desvio padrão das requisições são utilizados apenas os valores diferentes de zero. Essa abordagem é adotada a fim de se evitar distorções em relação a resolvidores que ficam grandes períodos do dia inativos e, em alguns momentos, geram uma quantidade muito grande de consultas. Ou seja, apenas são utilizados para cálculo os valores em relação aos períodos em que os resolvidores de fato estavam enviando requisições. Essas novas informações geradas são acrescentadas às outras informações já existentes, sobre os resolvidores, no banco de dados.

3.6 ATRIBUTOS PARA CLASSIFICAÇÃO DOS RESOLVEDORES

Para que seja possível a realização da clusterização, são necessários atributos. Essa escolha dos atributos influencia diretamente na qualidade do agrupamento final, e, portanto, no próprio resultado pretendido. Conforme já tratado anteriormente, não são coletadas muitas informações a respeito das requisições em si e, em virtude disso, o processo de escolha de atributos que não fossem redundantes foi criterioso. Seguindo essa linha de raciocínio não foram escolhidos como atributos a quantidade total de requisições diárias e a quantidade de requisições por minuto, por exemplo, mas sim indicadores que não trouxessem tantas semelhanças e/ou relacionamentos diretos.

⁷O SHA256 corresponde a um algoritmo de *hash* que utiliza 256 bits para criptografia. Atualmente é considerado seguro sendo amplamente utilizado para funções de criptografia embora existam outros que forneçam uma segurança ainda maior quanto a sua resistência.

⁸O *scp* trata-se de um protocolo que permite transferir seguramente arquivos entre um local e um host remoto ou entre dois hosts remotos.

⁹MongoDB é um banco de dados NoSQL orientado a documentos. Ele utiliza documentos com estrutura semelhante a JSON e possui suporte a transações ACID

Assim sendo, elegeu-se como atributos a quantidade total de requisições feitas, a quantidade de minutos em que o resolvedor esteve ativo realizando requisições e a quantidade de picos que ele possui em sua distribuição de consultas ao longo do dia analisado.

Talvez o atributo mais indispensável seja a quantidade total de requisições feitas. Isso porque é evidente que, quanto mais requisições um resolvedor tiver realizado, maior a probabilidade de que tenha comportamento agressivo e maior o seu potencial de afetar negativamente o DNS. Contudo, este não é o único indicador de que um resolvedor esteja sendo abusivo. Até porque, se assim o fosse, bastaria estipular que um resolvedor com uma quantidade de consultas acima de um limite seria agressivo. É preciso recordar que resolvedores podem ter comportamentos muito diversos entre si, ainda que possuam ao final do dia a mesma quantidade de requisições feitas. Enquanto alguns poderão ter suas requisições distribuídas ao longo do dia, outros poderão ter todas as requisições agrupadas em um único momento e não ter realizado qualquer atividade no restante do dia.

É neste sentido que o segundo atributo escolhido trata da existência de atividade a cada minuto. Um resolvedor poderá ter atividade em apenas um minuto ou durante todos os 1440 minutos do dia analisado. Este indicador, em conjunto com os outros atributos permite identificar de que forma é o padrão de atividade de um dado resolvedor.

Por fim, resolvedores, ativos o tempo todo ou não, podem ter ou não picos se analisada a distribuição de suas consultas. Um pico pode ser entendido como um comportamento anormal e muito discrepante de um padrão observado ou estabelecido. Mas como exatamente compreender e estipular o que deverá ser considerado pico no cenário em tela?

Inicialmente, utilizou-se bibliotecas de identificação de anomalias a fim de determinar quais pontos poderiam ser considerados picos, contudo, cabe lembrar que os resolvedores possuem comportamentos muito diferentes entre si. Assim, a tentativa de utilização de bibliotecas para a identificação dos picos não foi muito eficaz. Dentre uma das bibliotecas que foi inicialmente adotada está a *adtk (Anomaly Detection Toolkit)*¹⁰ cujo fim se destina a detecção de anomalias de série temporal não supervisionada. Ela oferece um conjunto de detectores, transformadores e funções para processar e visualizar séries temporais e eventos de anomalia. Ao aplicar os diferentes detectores presentes no pacote *adtk*, verificava-se que o limite ótimo (*thresholds*) para uma quantidade de resolvedores ou perfil de resolvedores era inadequado para a outra parte deles. Assim, não se chegava a um cenário ideal em que os picos pudessem ser corretamente identificados independentemente da quantidade de consultas e da sua distribuição.

A partir disso, optou-se pelo emprego de uma estratégia própria de identificação de picos em que se o valor do dobro do desvio padrão das consultas de determinado resolvedor for menor que sua média, então serão considerados picos os valores cujo desvio padrão seja maior que quatro e em que a quantidade de requisições naquele minuto seja maior do que duas vezes a sua média. De outra forma, se o valor do dobro do desvio

¹⁰Disponível em <https://adtk.readthedocs.io/en/stable/>

padrão das consultas de determinado resolvedor for maior que sua média então serão considerados picos os valores cujo desvio padrão seja maior que quatro e em que a quantidade de requisições naquele minuto seja a média apurada somada de duas vezes o seu desvio padrão.

Esta estratégia garante que, nos casos em que o desvio padrão seja muito pequeno em relação à média, valores pouco superiores à média, ainda que várias vezes superiores ao desvio padrão, não sejam considerados picos e que em casos em que resolvedores tenham um desvio padrão alto não sejam considerados picos quaisquer valores que excedam razoavelmente sua média, mas não muitas vezes o desvio padrão. Em resumo, se o desvio padrão das consultas de um dado resolvedor agrupadas por minuto for pequeno será dada mais atenção aos valores que excedam duas vezes a média (já que não cabe a utilização do desvio padrão sendo ele tão baixo) e se, todavia, o valor do desvio padrão das consultas for alto, então pensar em valores que excedam sua média não é suficiente, sendo necessário também somar desvios padrão que excedam a média encontrada.

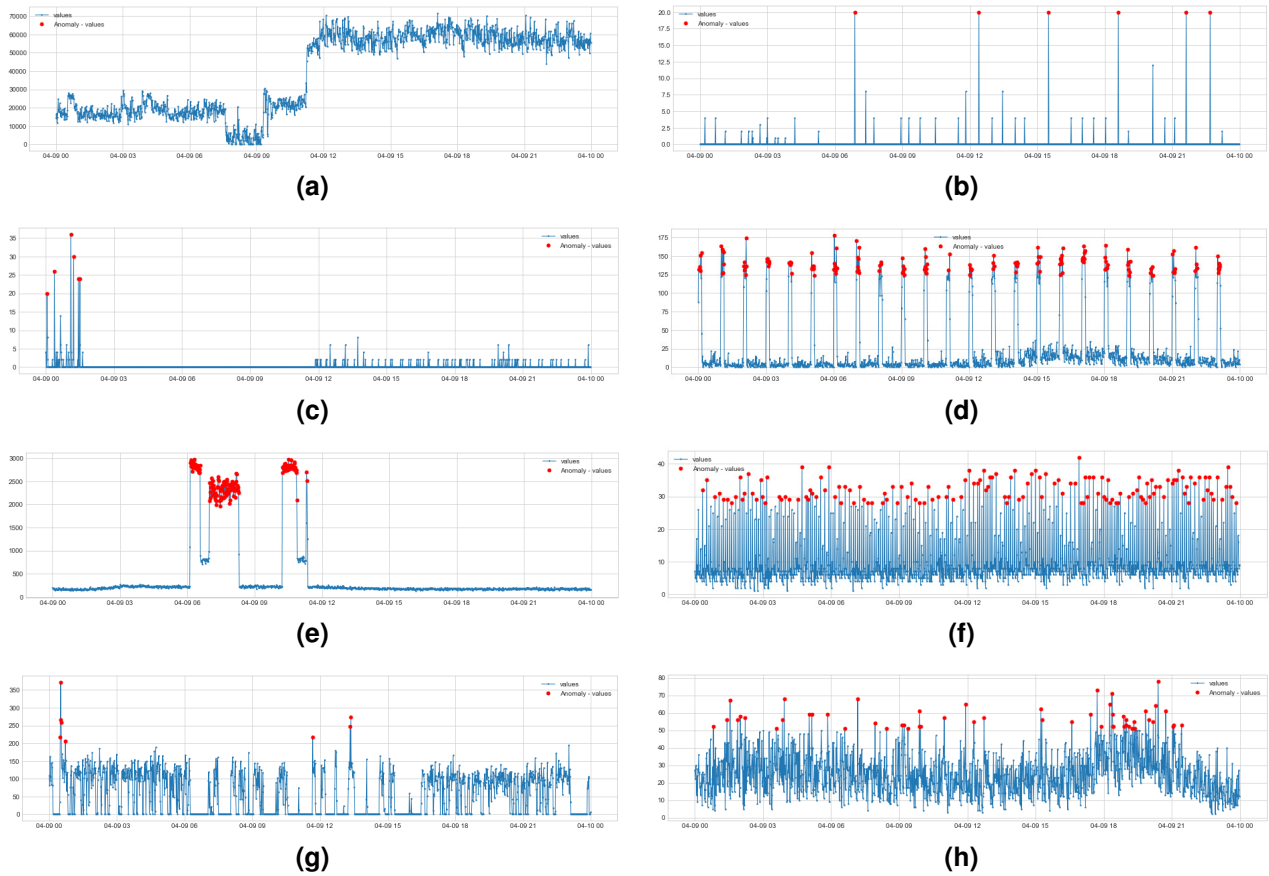
Cabe dizer que apenas foram considerados picos aqueles valores que excediam quatro vezes o valor de desvio padrão calculado para o resolvedor, a fim que de não fossem considerados como tendo picos aqueles resolvedores com quantia insignificante de consultas por minuto, mas que, por serem de valor tão baixo, ainda assim corresponderiam aos critérios para os casos de teste explanados anteriormente.

Encontrar uma tática que representasse corretamente os picos dos resolvedores foi desafiador contudo, parece que a estratégia proposta está adequada ao que se espera que é identificar aqueles momentos em que as consultas do resolvedor divergem muito de seu padrão e cujos valores são razoavelmente impactantes para serem considerados.

Conforme se pode ver da Figura 10, um resolvedor poderá ter um número variado de picos, partindo-se de zero. Os picos, por sua vez, poderão ser determinados de acordo com os critérios já explicados anteriormente, e que dependerá dos valores previamente calculados de média e desvio padrão das quantidades de requisições agrupadas a cada um minuto, considerando-se apenas os minutos em que tenha havido pelo menos uma requisição.

Dos três atributos, foi necessário realizar transformação logarítmica (\log_{10}) do 'quantidade de requisições' tendo em vista a extensão de valores entre resolvedores que menos e mais fizeram requisições. Em contrariedade ao atributo relacionado a minutos ativos, em que os resolvedores serão distribuídos em uma faixa de 1 a 1440, a quantidade de requisições comporta uma faixa de 1 até a quantidade total de requisições que um resolvedor tiver feito sozinho (quase 77 milhões em 2019 e mais de 160 milhões em 2020, por exemplo).

Assim, após a transformação do atributo de quantidade de requisições, passou-se a observar a distribuição de frequências retratada através da Figura 11. Como exemplo,



(a) - sem picos identificados. (b), (c), (d), (e) e (f) - picos identificados através da regra de que se o valor do dobro do desvio padrão das consultas de determinado resolvidor for menor que sua média, então serão considerados picos os valores cujo desvio padrão seja maior que quatro e em que a quantidade de requisições naquele minuto seja maior do que duas vezes a sua média. (g), (h) - picos identificados através da regra de que se o valor do dobro do desvio padrão das consultas de determinado resolvidor for maior que sua média então serão considerados picos os valores cujo desvio padrão seja maior que quatro e em que a quantidade de requisições naquele minuto seja a média apurada somada de duas vezes o seu desvio padrão.

Figura 10: Exemplos de resolvidores e picos identificados no DITL 2019. Fonte: Autora.

utilizou-se o DITL de 2019, contudo a distribuição segue o mesmo padrão nos diferentes *datasets*.

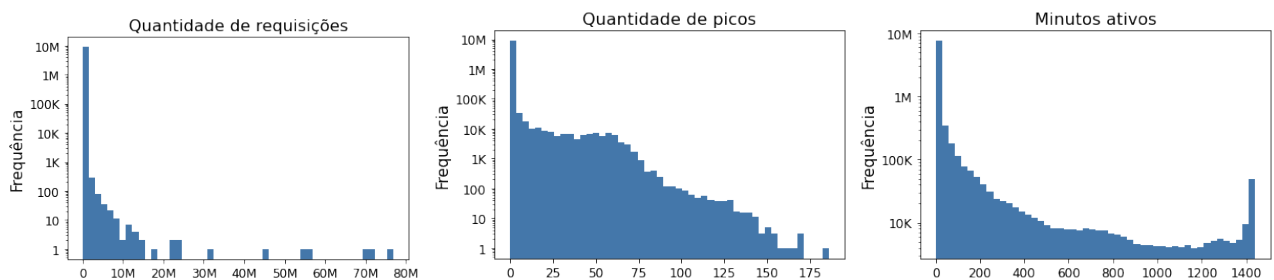


Figura 11: Distribuição de frequências de resolvidores por atributos no DITL 2019. Fonte: Autora.

A fim de facilitar a visualização e análise dos dados, utilizou-se escala logarítmica para representar os valores do eixo 'y' dos gráficos da Figura 11.

Como se pode ver da Figura 11, a grande maioria dos resolvedores fizeram uma pequena quantidade de requisições, possuem poucos ou nenhum pico e esteve fazendo requisições por poucos minutos durante o dia analisado. Ainda sem qualquer estudo e análise avançados é possível inferir que a grande maioria dos resolvedores a serem agrupados não serão abusivos.

Por fim, para determinar a qualidade dos atributos escolhidos, foi utilizado o coeficiente de Spearman, o qual mede a força e a direção da associação entre duas variáveis classificadas, através de uma relação monotônica [57]. A correlação está demonstrada através da Figura 12.

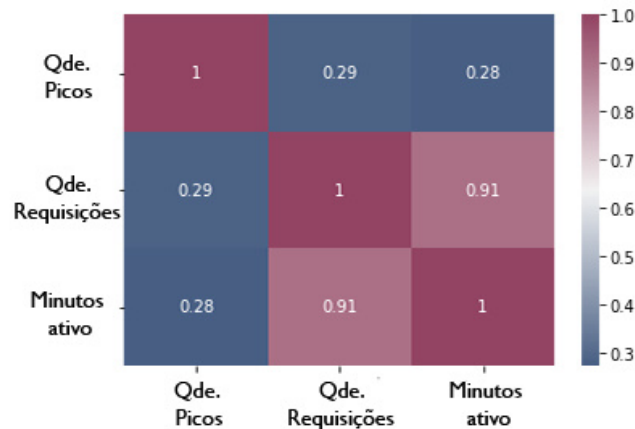


Figura 12: Correlação entre variáveis através do método Spearman. Fonte: Autora.

Segundo o método Spearman, a correlação pode variar de 1 a -1 sendo que valores mais próximos aos extremos indica uma maior correlação e valores mais próximos a zero uma menor correlação. O sinal, se positivo ou negativo, indica a direção da relação. Caso positivo, o aumento de uma variável implica no aumento da outra e, caso negativo, o aumento de uma variável implica na diminuição da outra.

Do que se pode ver, os atributos com maior relação entre si são o de quantidade de requisições feitas pelo resolvedor e a quantidade de minutos ativos (0,91%). Trata-se de uma relação forte e qual não influencia negativamente o modelo em virtude das outras correlações existentes, que, por serem fracas, permitem que os dados não sejam avaliados segundo critérios muito próximos. Essas outras relações (fracas) tratam-se das existentes entre quantidade de picos e minutos em que o resolvedor esteve ativo (0,28) e quantidade de requisições feitas e quantidades de picos de um resolvedor.

Uma vez determinados os atributos a serem utilizados no aprendizado de máquina, passou-se a tentativa de classificação dos resolvedores através do método de clusterização.

3.7 CLASSIFICAÇÃO DOS RESOLVEDORES ATRAVÉS DE GMM

Como já falado anteriormente, tentou-se utilizar como algoritmo de clusterização o K-Means, sendo este baseado em distâncias Euclidianas. Foram adotados como atributos os já citados, quais sejam, quantidade total de requisições feitas, quantidade de minutos em que o resolvedor esteve ativo realizando requisições e a quantidade de picos que ele possui em sua distribuição de consultas ao longo do dia analisado. Também, foi utilizada a técnica de transformação logarítmica dos dados referente ao atributo de quantidade de requisições. A quantidade de grupos foi determinada através da técnica de curva do cotovelo que busca calcular as distâncias até o centro do agrupamento a que elas pertencem buscando que a soma dos quadrados *intra-clusters* seja o mais próximo de zero possível.

De acordo com a resposta apontada pelo método da curva do cotovelo, as requisições foram agrupadas em três *clusters* e, com isso, observou-se que o K-Means não é a estratégia mais adequada para o cenário que se possui.

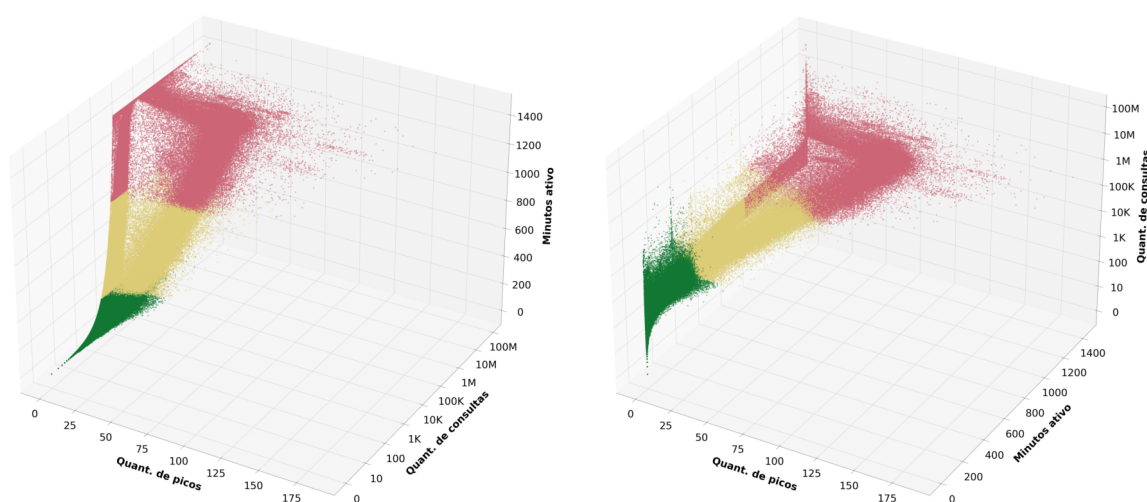


Figura 13: DITL 2019 agrupado por meio de K-Means. Fonte: Autora.

Conforme é possível perceber a partir da Figura 13, o K-Means apresentou um resultado em que uma linha reta foi traçada para separar os grupos, desconsiderando suas disposições e densidades. Tal fato observado é corroborado com as características próprias e já conhecidas desse algoritmo. O K-Means caracteriza-se por não funcionar bem em cenários em que os grupos não são circulares e são de tamanhos muito diferentes, além de não utilizar probabilidade para atribuição de dados a grupos [58].

Além do K-Means, outros métodos de clusterização foram considerados e descartados, tal como o DBSCAN que se caracteriza pela formação de grupos, baseado em densidade. Este método também se revelou desapropriado haja vista que o DBSCAN não é adequado para situações em que os *clusters* possuem densidades muito diferentes [59], o que verificou-se ser a situação nos conjuntos de dados utilizados.

Após estudo acerca de diferentes métodos de agrupamento, passou-se à utilização do método conhecido como *Gaussian Model Mixture* descrito no capítulo 2. A fim de melhorar o agrupamento, utilizou-se o inicializador k-means. Esse inicializador procura espalhar o conjunto inicial de centróides para que eles não fiquem muito próximos. Ele é conhecido por melhorar a qualidade dos ótimos locais e diminuir o tempo médio de execução [60].

A fim de que fosse determinada a quantidade ideal de grupos para clusterização, utilizou-se o coeficiente *silhouette* e o *score* BIC. Embora esses critérios não indiquem precisamente a quantidade de grupos a ser adotada, eles trazem uma estimativa de qualidade do modelo para um determinado conjunto de dados. O coeficiente *silhouette* considera a distância média entre uma amostra e todos os outros pontos no mesmo *cluster* e a distância média entre uma amostra e todos os outros pontos no aglomerado mais próximo [61]. Por sua vez o critério de informação Bayesiano (BIC) [62] baseia-se em função de verossimilhança e um valor mais baixo tende a indicar um melhor ajuste.

Após a utilização dos critérios BIC e coeficiente *silhouette* e de alguns testes para confirmar o número ideal de agrupamentos, adotou-se para a classificação dos resolvedores a formação de quatro diferentes grupos.

Além da adoção do inicializador k-means, e da indicação de quatro componentes, utilizou-se ainda como parâmetros, a opção padrão “*full*” como tipo de covariância (cada componente tem sua própria matriz de covariância geral) e o valor 1e-4 para limite de convergência um resolvidor poderá ter um número variado de picos, partindo-se de zero [60]. Os resultados obtidos a partir da classificação são apresentados no capítulo 4.

4. APRESENTAÇÃO E DISCUSSÃO DE RESULTADOS

Neste capítulo serão apresentados os resultados obtidos através da utilização da técnica de aprendizado de máquina, conhecida como clusterização, adotada para a classificação dos resolvedores, assim como serão discutidos os resultados obtidos no que se refere à finalidade e objetivo deste trabalho. Por fim, será apresentada proposta de identificação de resolvedores cujo comportamento seja potencialmente agressivo a partir de métricas extraídas dos resultados das classificações realizadas.

4.1 RESULTADOS DA CLASSIFICAÇÃO A PARTIR DO USO DO GMM

Conforme já descrito anteriormente, após estudos relacionados a estratégias de clusterização, adotou-se o GMM para agrupar os dados presentes nos diferentes anos de DITL em grupos. A partir de resultados do critério BIC e coeficiente *silhouette* os dados, independentemente do ano de DITL, foram agrupados em quatro diferentes grupos correspondentes a resolvedores não agressivos ou cuja agressividade é baixa, média ou alta.

4.1.1 DITL 2016

O DITL de 2016 caracteriza-se por ser o menor *dataset* dentre os analisados. Foram observados 7,3 milhões de resolvedores responsáveis por cerca de 7,1 bilhões de consultas para 11 diferentes letras de *root servers*.

Para fins de apresentação, representou-se através de gráfico todos os resolvedores presentes no DITL de 2016. Esses resolvedores estão distribuídos em um campo tridimensional cujos eixos se referem aos atributos quantidade de picos, quantidade de consultas realizadas e a quantidade de minutos em que estiveram ativos, ou seja, quantidade de minutos do dia analisado em que efetuaram pelo menos uma requisição a qualquer dos *roots servers*.

Conforme evidencia a Figura 14, existe uma grande quantidade de pontos azuis correspondentes aos resolvedores que possuem um comportamento tido como altamente agressivo. Muito embora em uma visualização inicial seria razoável acreditar que esses resolvedores estão presentes em maior quantidade no conjunto de dados classificado, ao contrário, referem-se a uma quantidade pequena de resolvedores se comparada à quantidade total de resolvedores presentes no DITL analisado. Isso ocorre porque os pontos encontram-se mais dispersos, possuindo valores mais variados para os três atributos utilizados. Por sua vez, os resolvedores tidos como não agressivos estão representados de

forma quase inexpressível no gráfico, correspondendo, entretanto, a mais da metade dos resolvedores.

Dessa forma e de acordo com gráfico em barra que visa representar as quantidades de requisições acumuladas por cada grupo e quantidade de resolvedores acumulados por cada grupo, pode-se inferir que uma pequena quantidade de resolvedores é responsável por uma grande quantidade de consultas (pontos azuis) e que uma grande quantidade de resolvedores é responsável por uma pequena quantidade de consultas (pontos verdes).

De fato, os resolvedores tidos como não agressivos correspondem a cerca de 62,4% do total de resolvedores existentes para o conjunto de dados e realizaram apenas 0,2% de todas as consultas analisadas. Os resolvedores com agressividade baixa correspondem a 29,7% do total de resolvedores e foram responsáveis por 2,9% do total das requisições. Já os resolvedores com agressividade média correspondem a 4,7% do total dos resolvedores, tendo sido responsáveis por 6,6% do total das requisições. Por fim, os resolvedores que se entende como agressivos correspondem a 3,2% do total de resolvedores e enviaram 90,3% do total das consultas.

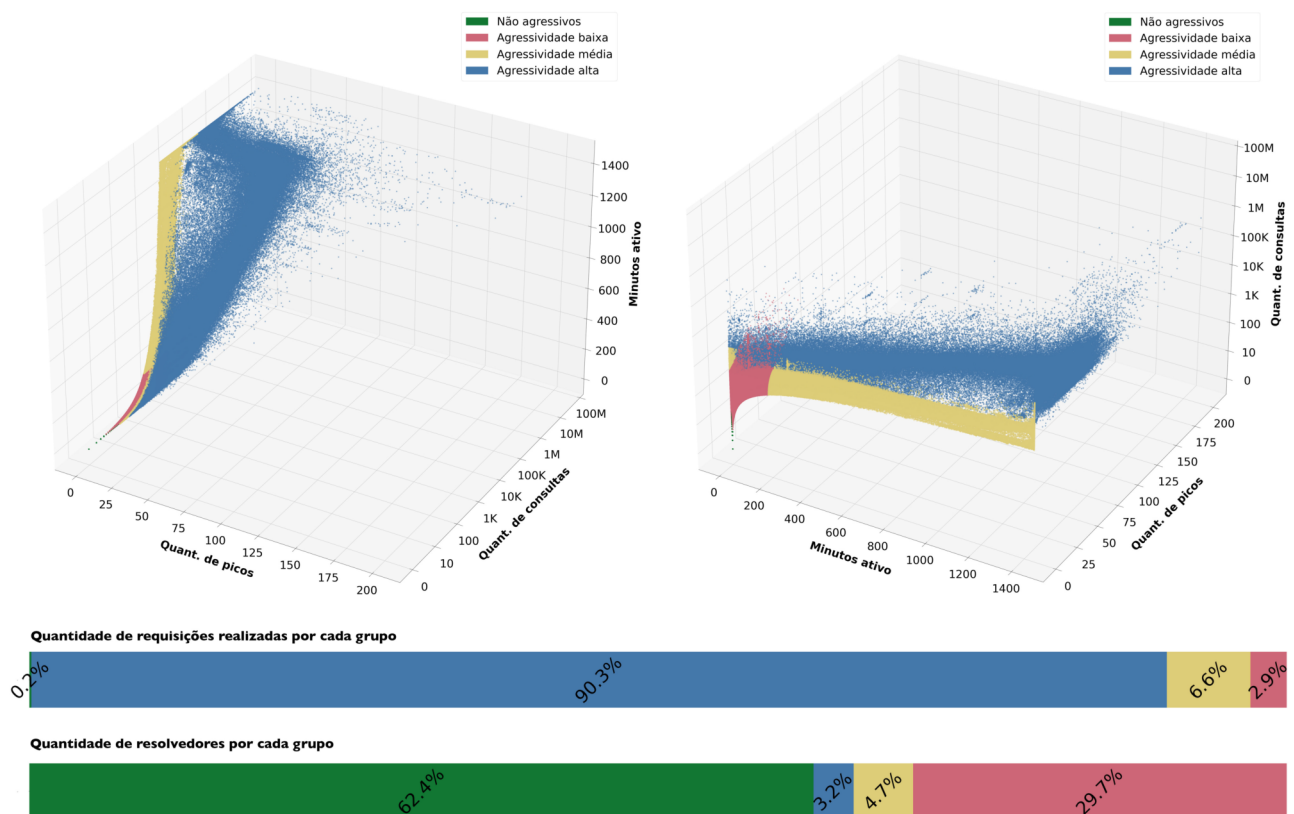


Figura 14: Clusterização para DITL de 2016. Fonte: Autora.

4.1.2 DITL 2017

No DITL de 2017 foi possível identificar 8,5 milhões de resolvedores os quais foram responsáveis pela realização de mais de 7,1 bilhões de consultas para 12 letras de servidores raiz.

A Figura 15 representa a distribuição desses resolvedores de acordo com os critérios de classificação já apresentados e também utilizados para o DITL de 2016.

Conforme se denota do gráfico, os resolvedores classificados como não agressivos correspondem a 58,9% do total de resolvedores e foram responsáveis por apenas 0,1% de todas as solicitações analisadas. Os resolvedores com agressividade baixa correspondem a 33,8% do total de resolvedores e realizaram 2,3% do total das requisições. Por sua vez, os resolvedores com agressividade média correspondem a 5,1% do total dos resolvedores, tendo sido responsáveis por 8% do total das requisições. Finalmente, os resolvedores tidos como agressivos correspondem a 2,2% do total de resolvedores e enviaram 89,6% do total das consultas.

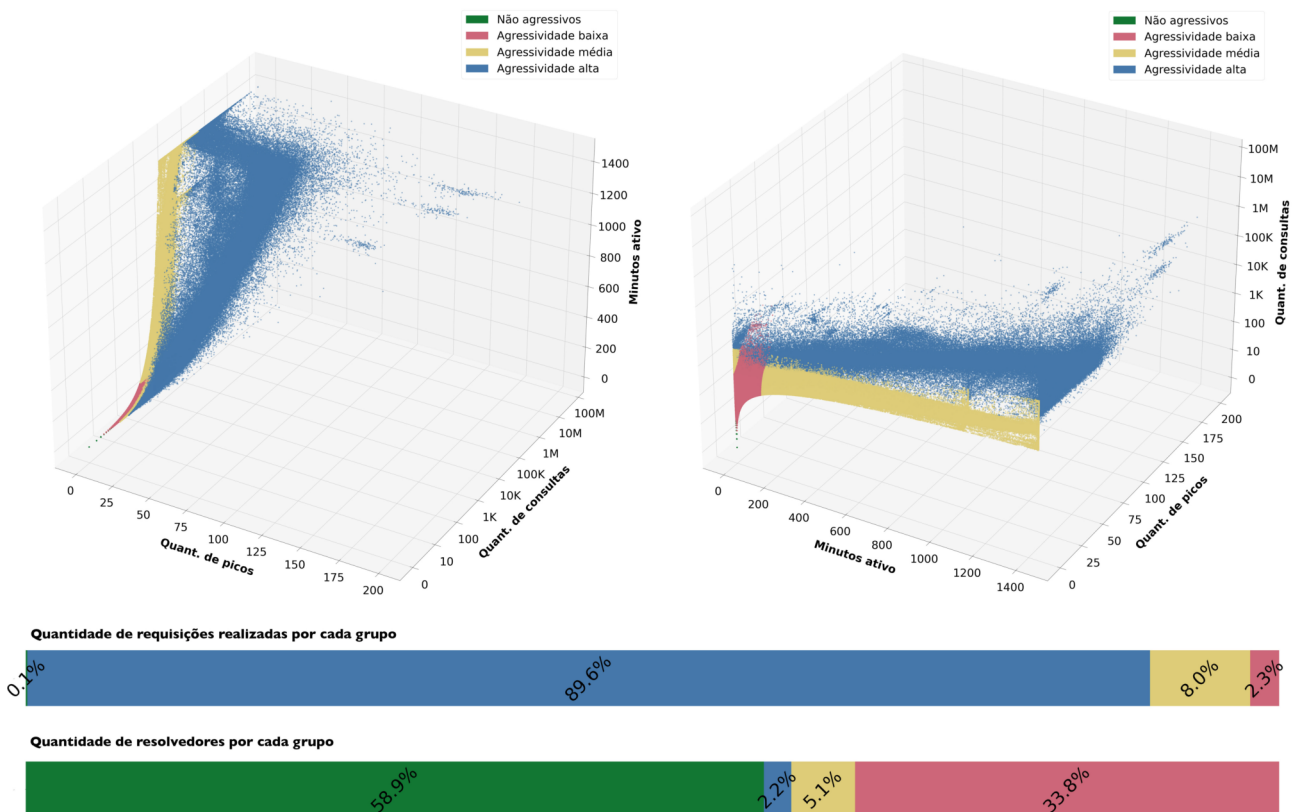


Figura 15: Clusterização para DITL de 2017. Fonte: Autora.

4.1.3 DITL 2018

O DITL de 2018 caracteriza-se por apresentar discreto aumento no número de resolvedores recursivos e de requisições que foram capturadas levando-se em consideração os critérios para coleta e utilização neste trabalho. Foram encontrados 9,6 milhões de resolvedores, os quais foram responsáveis pelo envio de 7,9 bilhões de consultas. Novamente, apenas o *root server* G não participou com o envio de dados para o conjunto de dados do ano analisado.

Este DITL também apresenta uma formação de grupos um pouco diferente em relação aos DITL anteriores. Isso porque, conforme representa a Figura 16, os pontos em azul ocupam uma parte maior do gráfico do que observado até então. Tal fato ocorre provavelmente em virtude de discrepâncias de valores de cada resolvedor, o que implica numa classificação diferente pelo algoritmo para atender essas dissimilaridades.

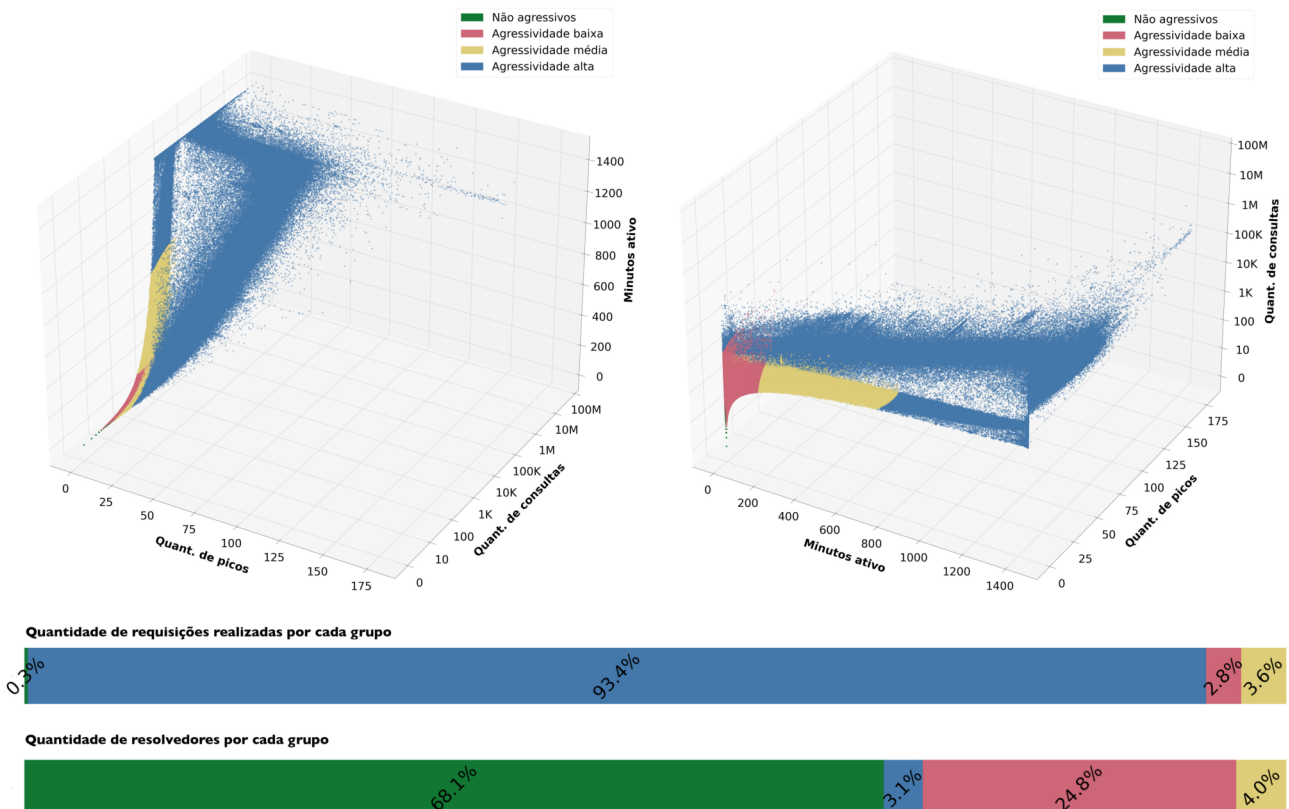


Figura 16: Clusterização para DITL de 2018. Fonte: Autora.

Dito isso, resolvedores considerados não agressivos correspondem a 68,1% e realizaram 0,3% das requisições. Resolvedores pouco agressivos correspondem a 24,8% do total de resolvedores e realizaram 2,8% das requisições. Resolvedores com agressividade média são 4% do total e fizeram 3,6% das requisições. E, por fim, os resolvedores cuja agressividade é mais alta correspondem a 3,1% do total do resolvedores e realizou 93,4% do total de consultas.

4.1.4 DITL 2019

Para o ano de 2019, foram apurados mais de 8,8 milhões de resolvedores os quais foram responsáveis pela realização de mais de 7,7 bilhões de consultas para 11 letras de servidores raiz (não há dados para os servidores raiz B e G). A ausência de dados para um dos *root servers* (letra B) talvez seja o motivo para que, em comparação com o ano anterior, tenha sido encontrado um menor número de resolvedores fazendo uso da estrutura DNS.

No que se refere a agrupamento, o gráfico para o ano de 2019 muito se assemelha dos gráficos obtidos para os anos de 2016 e 2017. Conforme é possível observar da Figura 17, resolvedores não agressivos foram responsáveis por enviar apenas 0,2% das consultas e eles são a grande maioria (65,4% do total).

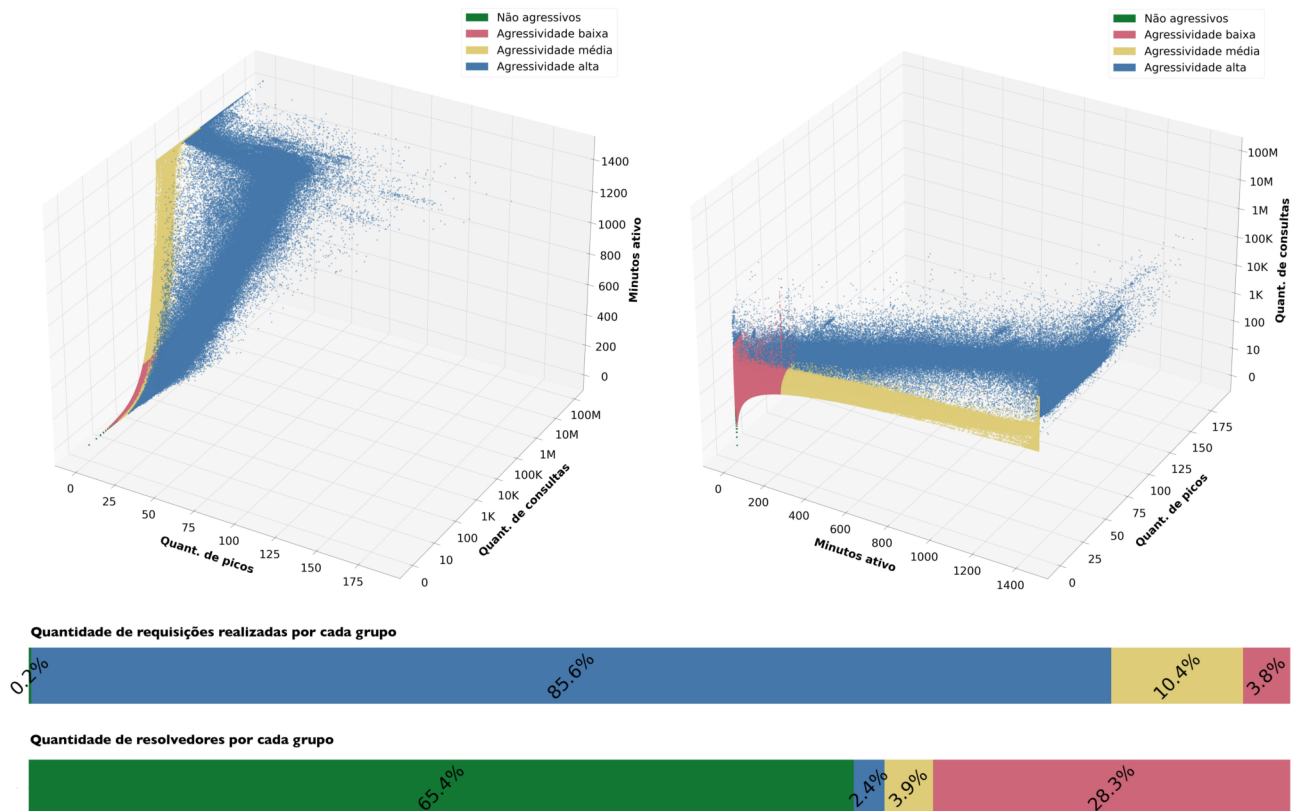


Figura 17: Clusterização para DITL de 2019. Fonte: Autora.

4.1.5 DITL 2020

No DITL de 2020, de acordo com os critérios utilizados, foi possível apurar a existência de mais de 12 milhões de resolvedores que juntos realizaram cerca de 8,7 bilhões de consultas para 12 letras de servidores raiz. Apesar de estarem presentes algumas consultas para o B-*root* (271 delas), é possível afirmar que essas poucas consultas não trazem qualquer implicação útil para a distribuição dos dados, sendo mais conveniente afirmar, in-

clusivo para fins estatísticos, que para 11 letras de *root server* foram realizadas mais de 8,7 bilhões de consultas.

Até o presente não há qualquer informação prestada pelo DNS-OARC que permita compreender porque requisições recebidas pelo B-root não se encontram no conjunto de dados de 2020. Como é realizado processo de limpeza sobre todos os dados coletados antes de sua disponibilização aos pesquisadores, é possível que dados eventualmente coletados tenham sido descartados. Também, é possível que o B-root quase não tenha participado com coleta de dados para esse ano.

É imprescindível ressaltar ainda que o ano de 2020 foi marcado pela continuidade de pandemia de corona vírus iniciada no final de 2019 (COVID-19), que tem impactado sobremaneira a economia e política mundial. Esta pandemia trouxe diversos reflexos em práticas culturais e implicações nas relações sociais que, inevitavelmente, refletiram em um aumento no volume de tráfego na Internet.

Coletas e pesquisas futuras permitirão identificar se o aumento da quantidade de resolvedores e do número de consultas realizadas pelos resolvedores que mais fizeram requisições pode ter ocorrido em decorrência dos efeitos provocados por esta emergência sanitária.

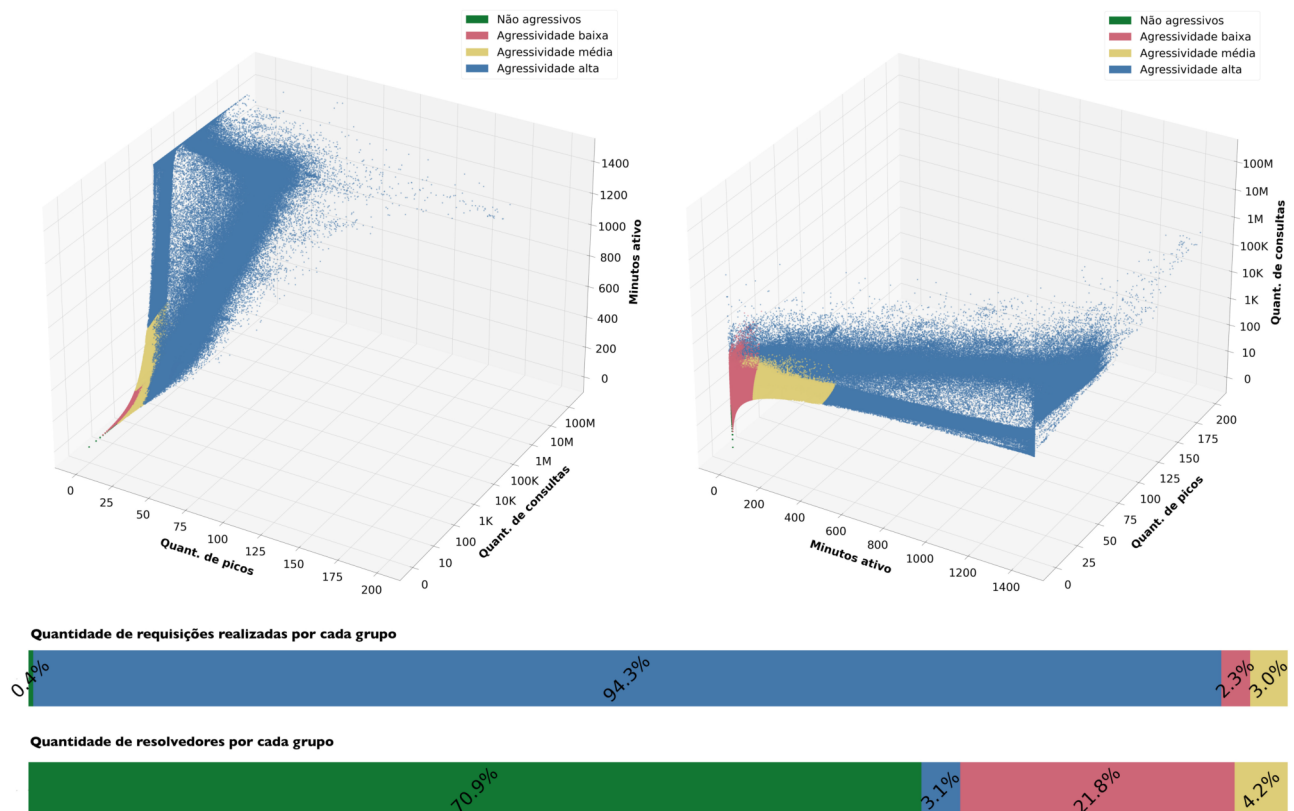


Figura 18: Clusterização para DITL de 2020. Fonte: Autora.

No que tange à classificação dos dados extraídos do DITL 2020, o agrupamento das consultas obtido é visivelmente semelhante ao do DITL de 2018, no qual os pontos em

azul ocuparam uma parte maior do gráfico do que observado até então. Apurou-se que os resolvedores não agressivos correspondem a 70,9% do total dos resolvedores e foram responsáveis por 0,4% do total das requisições. Os resolvedores pouco agressivos correspondem a 21,8% do total de resolvedores e realizaram 2,3% do total das requisições. Por sua vez, os resolvedores com agressividade média correspondem a 4,2% do total dos resolvedores, tendo sido responsáveis por 3% do total das requisições. Finalmente, os resolvedores tidos como agressivos correspondem a 3,1% do total de resolvedores e enviaram 94,3% do total das consultas.

4.2 ANÁLISE DOS RESULTADOS OBTIDOS

A fim de facilitar comparações a serem realizadas entre os resultados obtidos com a classificação nos diferentes DITLs, apresenta-se novamente tabela com valores agregados para cada ano, agora incluindo variações, a qual pode ser conferida a seguir.

Tabela 10: Informações da coleta de dados nos servidores de nomes DNS. Fonte: Autora.

Data utilizada	Qtd. de consultas	Variação	Qtd. de resolvedores únicos	Variação
05/04/2016	7.163.357.401	—	7.396.026	—
12/04/2017	7.157.607.184	-5.750.217	8.527.792	+1.131.766
11/04/2018	7.904.917.521	+747.310.337	9.603.349	+1.075.557
09/04/2019	7.746.604.264	-158.313.257	8.859.099	-744.250
06/05/2020	8.702.203.719	+955.599.455	12.332.252	+3.473.153

Nota: A variação foi calculada levando-se em consideração a diferença de um ano para o ano imediatamente anterior

Da mesma forma, apresenta-se a Figura 19 que compila os percentuais de quantidade de resolvedores presentes e de consultas realizadas por cada um dos quatro grupos resultantes da classificação através de GMM.

Conforme se pode notar da Figura 19, as distribuições dos grupos, seja em relação à quantidade de resolvedores, seja em relação à quantidade de consultas possui certa semelhança, principalmente entre os anos de 2018 e 2020, o que implica na própria semelhança observada nos gráficos para estes dois anos em contraste aos outros três.

Contudo, pensando-se na diferença observada no padrão de formação de alguns grupos de 2018 e 2020 em contraste com os outros três anos (grupo azul assumindo pontos que antes eram pertencentes ao grupo amarelo), é possível presumir que tal peculiaridade ocorre em virtude das características dos destes próprios *datasets*. A Tabela 11 apresenta as medianas para minutos em que os resolvedores estiveram ativos, medianas das quantidades de picos e medianas das quantidades de consultas para cada grupo e, conseqüentemente, para cada DITL.

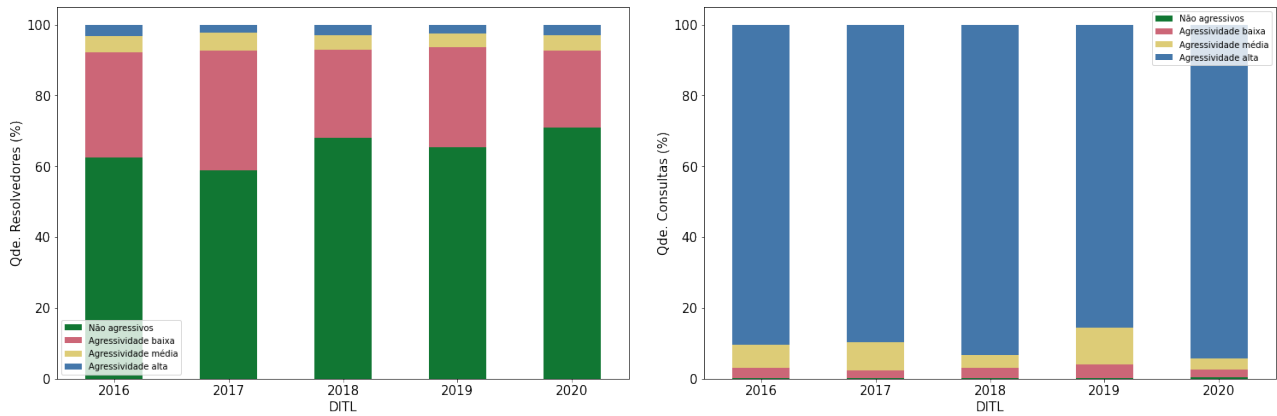


Figura 19: Comparação da quantidade percentual de resolvedores e consultas presentes em cada grupo, pós classificação, para cada DITL. Fonte: Autora.

Tabela 11: Medianas dos atributos para cada grupo classificado e DITL. Fonte: Autora.

	2016				2017				2018				2019				2020			
	não agr.	agr. baixa	agr. méd.	agr. alta	não agr.	agr. baixa	agr. méd.	agr. alta	não agr.	agr. baixa	agr. méd.	agr. alta	não agr.	agr. baixa	agr. méd.	agr. alta	não agr.	agr. baixa	agr. méd.	agr. alta
min. ativo.	1	15	309	367	1	9	258	487	1	14	219	713	1	15	406	362	1	8	107	760
qtd. picos	0	0	0	11	0	0	0	13	0	0	0	11	0	0	0	13	0	0	1	7
qtd. req.	2	35	850	2.508	1	21	640	3.924	2	40	501	3.880	1	43	990	2.880	2	36	355	3.077

Embora não seja possível determinar com absoluta certeza quais os limiares para cada atributo ou relações existentes entre os elementos dos *datasets* sejam responsáveis diretamente pela diferença de distribuição notada nos DITL de 2018 e 2020 em relação ao padrão dos demais, é interessante notar que, conforme Tabela 11, justamente os DITL 2018 e 2020 são aqueles que tem uma maior diferença nas medianas de minuto ativo para os grupos cuja agressividade é média ou alta. No ano de 2018, a mediana de minutos ativos para resolvedores com agressividade média ficou em 219 e a mediana de minutos ativos para resolvedores com agressividade alta ficou em 713. Já para o DITL de 2020, a mediana de minutos ativos para resolvedores com agressividade média ficou em 107 e a mediana de minutos ativos para resolvedores com agressividade alta ficou em 760. Ou seja, a diferença visualmente observável entre os gráficos fica clara também quando se analisam as medianas.

Em que pese tenha havido alguma diferença entre a distribuição dos elementos pertencentes aos grupos com agressividade média e alta, não se pode concluir que a classificação está incorreta. Isso porque os dados não são rotulados e não se trata de uso de algoritmo de aprendizagem supervisionada e também porque é justamente o GMM que determina que parte da população pertence a cada grupo, considerando-se as probabilidades calculadas para cada atributo, considerando as particularidades de cada conjunto de dados.

4.2.1 Análise dos resolvedores com agressividade alta

Após a classificação realizada, ficou claro que, embora grande parte dos resolvedores possuam comportamento agressivos, haja vista que fazem mais consultas diárias do que o esperado caso considerassem os valores de TTL recomendados, há aqueles resolvedores cujo potencial de nocividade ao DNS é ainda maior. Estes resolvedores, conforme já visto, são representados nos gráficos pela cor azul e correspondem a uma faixa de 2,19 a 3,18 % do total de resolvedores e 85,62 a 94,35% do total de consultas, para os 5 DITL analisados.

Ainda que o estudo de todos os resolvedores que de alguma forma tenham comportamento abusivo seja pertinente e necessário, neste trabalho, em virtude de escopo e tempo hábil para desenvolvimento das atividades, foi necessário dar prioridade ao estudo dos resolvedores classificados como os mais agressivos.

Dito isso e a fim de possibilitar o estudo da presença e do impacto desses resolvedores ao longo dos DITLs, apresenta-se a Tabela 12 que consolida algumas informações ao longo dos anos. Cabe comentar que, nesta subseção, para fins de simplificação, serão denominados “resolvedores agressivos” os resolvedores classificados em grupo “agressividade alta” e identificados pela cor azul. Da mesma forma, serão denominados “top resolvedores agressivos” aqueles resolvedores agressivos (pertencentes ao grupo azul) que tenham contribuído para a realização de cerca de 99% das consultas realizadas pelo seu próprio grupo (azul) para cada ano de DITL.

Tabela 12: Informações sobre resolvedores agressivos. Fonte: Autora.

DITL	Qtd. resolv. agr.	Qtd. consultas	Qtd. top resolv. agr.	Qtd. consultas top resolv.
2016	235.080	6.467.960.553	135.357	6.403.282.439
2017	186.630	6.411.117.081	107.382	6.347.007.163
2018	294.666	7.380.058.765	207.298	7.306.259.726
2019	215.139	6.632.439.789	123.625	6.566.115.555
2020	381.231	8.210.480.311	295.170	8.128.375.921

Sobre esses resolvedores top agressivos cabe ainda explicitar que, eles correspondem a 606.298 IP diferentes. Há, portanto, resolvedores que figuram entre os top resolvedores agressivos em diferentes DITL.

As Tabelas 13 e 14 a seguir apresentam quantos resolvedores participaram dos DITL e quantos são comuns entre um DITL e outro.

Tabela 13: Top resolvedores agressivos comuns para cada ano de DITL. Fonte: Autora.

Qtd. DITL	Qtd. Resolvedores	Percentual
1	456.508	75
2	83.310	14
3	34.607	6
4	17.482	3
5	14.391	2
TOTAL	606.298	100

É claro que a grande parte dos top resolvedores agressivos esteve presente em apenas um dos DITLs. Na verdade 75% deles esteve presente em apenas um DITL e cerca de apenas 2% esteve presente nos 5 DITL. Isso corresponde a 14 mil resolvedores dos 606 mil analisados.

A Tabela 14, por sua vez, traz o total de top resolvedores agressivos por DITL e realiza o cruzamento entre esses diferentes DITL a fim de, para cada conjunto de dois anos, apresentar a quantidade bruta comum entre eles.

Tabela 14: Quantidades de top resolvedores agressivos comuns entre diferentes anos de DITL. Fonte: Autora.

		DITL 2016	DITL 2017	DITL 2018	DITL 2019	DITL 2020
	Qtd. resolv.	135.357	107.382	207.298	123.625	295.170
DITL 2016	135.357	x	52.384	49.761	24.301	27.640
DITL 2017	107.382	52.384	x	59.198	29.559	30.390
DITL 2018	207.298	49.761	59.198	x	50.994	61.068
DITL 2019	123.625	24.301	29.559	50.994	x	50.638
DITL 2020	295.170	27.640	30.390	61.068	50.638	x

A fim de trazer um comparativo visual à Tabela 14, com a Figura 20 pretende-se mostrar de forma mais clara a relação de repetição dos resolvedores top agressivos ao longo dos DITL.

Da Figura 20 é possível inferir que os resolvedores do DITL de 2020 pouco são os mesmos daqueles do DITL de 2016, mas, a medida em que os anos passam, cada vez passam a ser mais comuns. Assim, passam de uma taxa de 20% dos resolvedores de DITL de 2016 para 41% dos resolvedores do DITL de 2019. Já os resolvedores do DITL de 2016, por exemplo, estão cada vez menos presentes nos próximos DITL. Passam de uma taxa de 49% dos resolvedores do DITL de 2017 para cerca de 9% dos resolvedores do DITL de 2020.

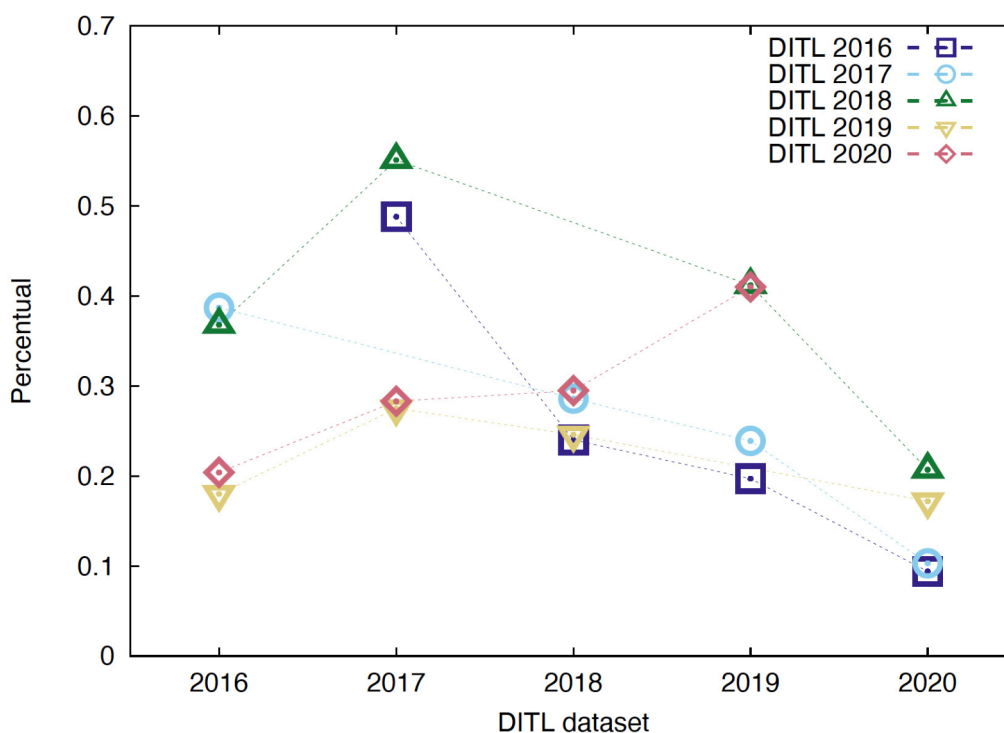


Figura 20: Top resolvedores agressivos existentes em comum entre diferentes anos de DITL. Fonte: Autora.

Por fim, na Tabela 15 são trazidos os top 20 resolvedores agressivos, considerando-se todos os anos de DITL analisados, agrupados pelo seu IP (anonimizado) e ordenados pela quantidade total de consultas que realizaram. Apresenta-se ainda a média dos minutos em que esteve ativo e a média de quantidade de seus picos.

É possível perceber que, até a posição 10, constam apenas resolvedores que participaram de um ou dois DITL. Ainda que existam 14 mil resolvedores top agressivos que estejam presentes nos 5 DITLs analisados, eles não fizeram consultas o suficiente para superar os resolvedores presentes nas primeiras posições (e participantes de poucos DITL).

Pode-se notar ainda que a média de minutos ativos desses resolvedores fica muito próxima ao valor máximo (1440) e que eles tendem a possuir entre 22 e 65 picos durante o dia. Exceção importante que pode ser observada é a do resolvedor na posição 15, o qual não teve picos, contudo ainda figure entre os top agressivos. Isso ocorre porque a agressividade é analisada através da relação de diferentes atributos. Não basta ter apenas uma quantidade alta de minutos ativo (em que esteve enviando requisições) ou mesmo muitos picos ao longo do dia, por exemplo. É necessário considerar a interação entre esses diferentes atributos para que se possa considerar um resolvedor como agressivo ou não.

Outro fator importante que deve ser relatado é a questão de que serviços de resolução de nomes podem ter seus endereços IP alterados ao longo dos anos. Com os dados que foram capturados dos *datasets* DITL disponibilizados pelo DNS-OARC não foi possível identificar resolvedores que podem ser considerados os mesmos, embora tenham tido

Tabela 15: Top resolvedores agressivos ordenados por quantidade de consultas realizadas.
 Fonte: Autora.

Pos.	Resolvedor	Qtd. consultas	Qtd. DITL	Méd. min. ativo	Méd. qtd. picos
1	1fa91e4288...	165.141.567	2	1440	49,5
2	cd874235db...	148.988.287	2	1440	50
3	2f9f03ce76...	104.284.889	2	1440	30
4	3e6dda028a...	81.789.990	2	1440	61
5	1d7a101219...	78.937.904	2	1440	64,5
6	773c5dfed6...	77.565.700	2	1440	42,5
7	ce34557259...	74.186.836	1	1440	54
8	48fc96aeaa...	66.915.273	2	1409,5	33
9	fb37ea59c5...	60.353.750	2	1440	22,5
10	189f97c87a...	58.804.971	2	1440	54,5
11	d1b295b7f2...	57.594.612	5	1440	32,8
12	442d66126f...	51.721.421	5	1147,6	38,8
13	d2b794b07c...	51.719.761	5	1440	38,2
14	3a347e1eda...	51.609.632	5	1440	39
15	8209c231d3...	50.579.896	3	1440	0
16	900fff422a...	50.425.732	5	1440	37,2
17	0769ba19a8...	49.529.549	5	1440	45,8
18	a8b5ec5d3b...	49.420.739	5	1437,6	33,2
19	6ef70e4323...	47.743.422	5	1432,2	45,4
20	80752ff656...	47.691.296	5	1439,4	41

seu endereço de origem alterado. Contudo, o que se pode esperar da análise feita nesta subseção é a certeza quanto aos patamares mínimos identificados. Ou seja, no mínimo os resolvedores apontados participaram daqueles *datasets*, embora possam ter participado de mais e, no mínimo realizaram as quantidades de consultas apontadas, embora possam ter realizado ainda mais, caso considerados outros endereços IP que vieram a lhes identificar em dado momento.

5. CONCLUSÃO

Pretendeu-se com o presente trabalho o desenvolvimento de uma metodologia que permitisse classificar resolvedores recursivos de acordo com o seu comportamento quanto ao envio de consultas aos servidores raiz do DNS. Além da classificação, objetivou-se ainda a identificação e a quantificação desses resolvedores, de forma que fosse possível comparar diferentes cenários de acordo com cada conjunto de dados do projeto DITL da OARC.

Para a realização da classificação, foi utilizado o algoritmo de aprendizagem não supervisionada GMM, o qual permitiu a clusterização dos dados de resolvedores em diferentes grupos de acordo com valores de seus atributos. Após a tarefa de agrupamento foi possível notar que, ano após ano, os grupos mantiveram semelhantes características e medianas para os atributos considerados, o que permitiu concluir que o modelo utilizado foi o adequado para a resolução do problema de pesquisa. As pequenas diferenças observadas, principalmente plotagem dos grupos referentes aos anos de 2018 e 2020, deve-se ao fato de que os dados não são rotulados, sendo tarefa do GMM determinar que parte da população pertence a cada grupo, considerando as probabilidades calculadas para os atributos, particulares a cada conjunto de dados.

A contribuição deste trabalho não se resume apenas ao método que permite classificar os resolvedores, mas, principalmente, a todo o conhecimento que foi obtido para que se pudesse escolher o algoritmo adequado para resolução do problema, qual parametrização utilizar e quais atributos adotar a fim de se chegar a um resultado satisfatório. Além disso, a mineração, tratamento e entendimento acerca dos dados coletados dos servidores OARC-DNS permitiu observar de que forma as consultas coletadas estão distribuídas ao longo dos diferentes anos, para os diversos *root servers*, e também a distribuição dessas mesmas consultas de acordo com os diferentes tipos de recursos solicitados ou TLD requisitados.

Por se tratar de uma primeira iniciativa de classificação de comportamento de resolvedores perante servidores raiz esse trabalho não se trata de um estudo fechado e que exaure todos os pontos e resultados esperados. Com certeza há muito o que se evoluir a partir do que foi alcançado, ainda que os resultados obtidos sejam um bom ponto de partida para estudos mais avançados.

5.1 LIMITAÇÕES E DISCUSSÕES

Para o estudo, foram utilizados *datasets* relativos ao projeto DITL da DNS-OARC. Apesar de ser um projeto consolidado e reconhecido pela sua importância, ele objetiva

a coleta e disponibilização de dados de apenas 48 horas por ano e este fato deve ser considerado uma importante limitação na medida em que não se têm disponíveis dados de outros dias. Ainda assim, vale lembrar que os dias coletados tendem a representar um dia “normal” de vida do DNS. São escolhidos dias em que não tenha havido nenhum evento significativo que impacte os dados que serão colocados à disposição dos pesquisadores. É em virtude disso que se entende que o período coletado reflete satisfatoriamente a realidade do tráfego das consultas recebidas pelos *root servers*.

Há que se mencionar ainda as limitações existentes por se trabalhar com uma grande quantidade de dados. Mesmo após extração, mineração e classificação dos dados obtidos através do projeto DITL é o grande volume de requisições. Ainda que todas as transformações destes dados e que toda a criação de estatística possível tenha se dado em servidores da DNS-OARC, os quais possuem razoável poder computacional, trabalhar com dados compilados continua sendo uma tarefa custosa computacionalmente. Em virtude disso, deduz-se que trabalhar com os dados em *real time* ou com prazo de resposta curto é uma tarefa improvável para um operador de *root server*, ainda que este considere requisições apenas da letra de servidor que opera. Isso porque, conforme visto, na média foram encontradas cerca de 800 milhões de consultas para cada letra de servidor, considerando-se que foram coletadas apenas requisições com registro de recurso 1, 2, 28 e 43 e solicitação domínios com TDL com, cn e nl. Caso o operador deseje incluir todos os tipos de requisições e TLDs em suas análises, o número de consultas a serem analisadas será ainda maior.

No entanto, através de teste verificou-se que com uma amostra de um DITL é possível obter, proporcionalmente, as mesmas distribuições de elementos entre os grupos formados e semelhantes valores de média e medianas para os atributos utilizados. Ou seja, dependendo do objetivo pretendido com a classificação, uma amostra dos dados poderá ser suficiente para que sejam estabelecidos valores parâmetros para dado DITL, o que permitiria a classificação de resolvedores em tempo muito inferior ao que seria necessário caso se levasse em consideração o conjunto de dados de forma integral.

Por exemplo, para o DITL de 2016 foram coletadas 30% das requisições a título de amostra, as quais foram classificadas através de GMM. Um comparativo dos gráficos da classificação utilizando todas as requisições de 2016 e da classificação usando apenas 30% delas pode ser visualizada através da Figura 21.

Para ambas as classificações, os percentuais de quantidade de resolvedores em cada grupo foram praticamente os mesmos e situação semelhante pode ser observada para os valores de mediana das requisições considerando-se os atributos quantidade de picos, quantidade de minutos em que o resolvedor esteve ativo e quantidade de requisições feitas. A Tabela 16 apresenta as diferenças entre as duas classificações.

Chegando-se à estimativa de que 30% dos dados resultam em uma qualificação proporcionalmente similar a de utilização de todos os dados, em teste posterior dividiu-se

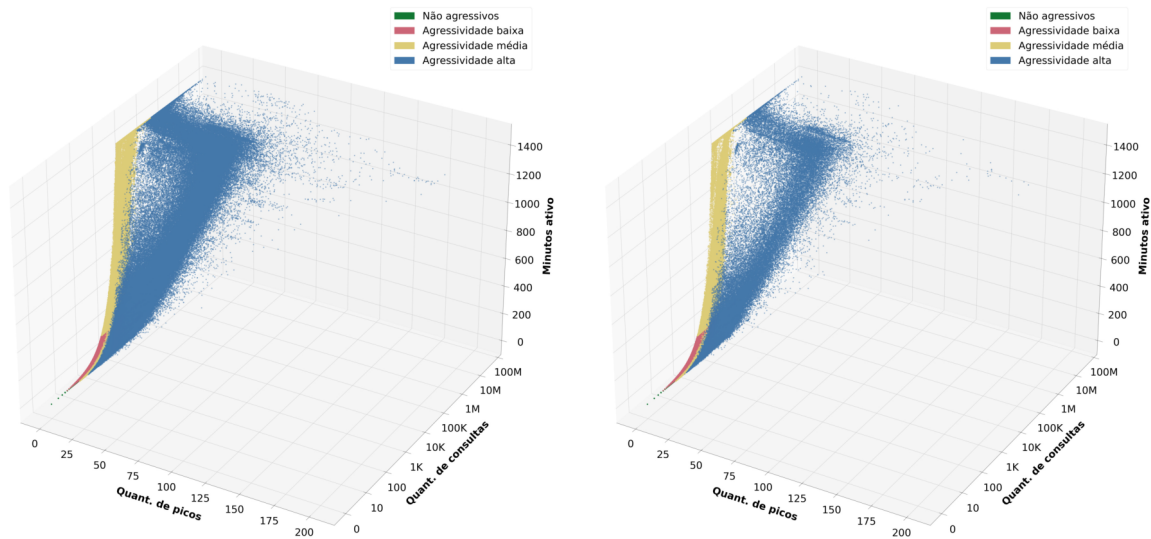


Figura 21: Comparativo entre classificações usando todo o conjunto (esq.) e parte do conjunto de dados (dir.) de 2016. Fonte: Autora.

Tabela 16: Estatísticas para o DITL 2016 - integralidade e amostra. Fonte: Autora.

	DITL 2016 (completo)					DITL 2016 (amostra)				
	percentual		mediana			percentual		mediana		
	Qtd. resolv.	Qtd. req.	Min. ativo	Qtd. picos	Qtd. req.	Qtd. resolv.	Qtd. req.	Min. ativo	Qtd. picos	Qtd. req.
Não agr.	62,37	0,18	1	0	2	62,39	0,17	1	0	2
Agr. baixa	29,71	2,89	15	0	35	29,70	2,72	15	0	35
Agr. méd	4,73	6,64	309	0	850	4,73	6,27	307	0	844
Agr. alta	3,18	90,29	367	11	2.508	3,18	90,85	367	11	2.502

o *dataset* de 2016 em conjunto de treino e conjunto de teste. Utilizou-se o resultado da clusterização dessa parte dos dados como rótulo em algoritmo de aprendizado supervisionado. Optou-se pelo *k-nearest neighbors* (kNN) tendo em vista ser este um intuitivo, simples, porém poderoso algoritmo de classificação.

Após treinamento e classificação com kNN, obteve-se uma taxa de acurácia de 0.99959 nas predições. Ou seja, com uma acurácia tão elevada é possível concluir que uma parte do conjunto de dados é suficiente para treinar o modelo e realizar as predições adequadamente, o que ajuda a contornar a dificuldade em torno do tempo necessário para realizar o agrupamento dos dados se utilizado todo o conjunto de dados.

É claro que a classificação com o kNN foi um teste inicial, ainda assim, aponta para uma solução válida a qual poderá ser validada ou mesmo testada a partir de outros algoritmos de classificação tais como árvores de decisão, SVM, Naive Bayes e outros citados no Capítulo 2.

5.2 DIREÇÕES FUTURAS

Este trabalho teve como um de seus objetivos realizar uma classificação inicial dos resolvedores procurando compreender de que forma estão distribuídos e em que quantidades compõem grupos relacionados aos seus comportamentos. Este estudo, contudo, não foi exaustivo e muito ainda pode ser feito a partir dos resultados que foram encontrados.

Partindo-se agora do pressuposto de que há um método determinado para realizar a classificação de resolvedores e que estes estão classificados de acordo com o grau de agressividade em seu comportamento, é possível concentrar esforços em identificar padrões que expliquem as causas de mau comportamento, uma vez que já são conhecidos os resolvedores “mais agressivos”.

Estudos conduzidos a partir dos resultados já obtidos permitirão compreender as causas para o mau comportamento dos resolvedores, como, por exemplo, memória cheia, *packet filtering*, configurações incorretas, entre outros. Desta forma, além da compreensão sobre quem são os resolvedores mais agressivos, será possível também determinar a melhor forma de agir em relação a eles, seja tomando medidas de contenção, seja realizando contato com seus responsáveis de forma a minimizar o impacto que os mesmos possam causar no DNS.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BROWNLEE, N.; CLAFFY, K.; NEMETH, E. DNS measurements at a root server. v. 3, p. 1672–1676 vol.3, 2001. Disponível em: <https://ieeexplore.ieee.org/document/965864>.
- [2] MOCKAPETRIS, P. *Domain Names - Concepts and Facilities*. [S.l.], 1987. Disponível em: <http://www.rfc-editor.org/rfc/rfc1034.txt>. Acesso: Abril/2021.
- [3] SARAT, S.; PAPPAS, V.; TERZIS, A. On the Use of Anycast in DNS. In: *Proceedings of 15th International Conference on Computer Communications and Networks*. [S.l.: s.n.], 2006. p. 71–78. Disponível em: <https://ieeexplore.ieee.org/document/4067629>.
- [4] MOURA, G. C. M. et al. Cache me if you can: Effects of DNS Time-to-Live (extended). n. ISI-TR-734b, jul. 2019. Disponível em: https://ant.isi.edu/datasets/dns/#Moura19a_data.
- [5] KALAFUT, A. et al. Surveying dns wildcard usage among the good, the bad, and the ugly. In: JAJODIA, S.; ZHOU, J. (Ed.). *Security and Privacy in Communication Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. ISBN 978-3-642-16161-2. Disponível em: https://link.springer.com/chapter/10.1007/2F978-3-642-16161-2_26.
- [6] LIU, D.; HAO, S.; WANG, H. All your dns records point to us: Understanding the security threats of dangling dns records. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA: ACM, 2016. (CCS '16), p. 1414–1425. ISBN 978-1-4503-4139-4. Disponível em: <http://doi.acm.org/10.1145/2976749.2978387>.
- [7] ZHANG, M. et al. How dns misnaming distorts internet topology mapping. In: *Proceedings of the Annual Conference on USENIX '06 Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2006. (ATEC '06), p. 34–34. Disponível em: <http://dl.acm.org/citation.cfm?id=1267359.1267393>.
- [8] ALLMAN, M. Comments on dns robustness. In: *Proceedings of the Internet Measurement Conference 2018*. New York, NY, USA: ACM, 2018. (IMC '18), p. 84–90. ISBN 978-1-4503-5619-0. Disponível em: <http://doi.acm.org/10.1145/3278532.3278541>.
- [9] AL-DALKY, R.; RABINOVICH, M.; SCHOMP, K. A look at the ecs behavior of dns resolvers. In: *Proceedings of the Internet Measurement Conference*. New York, NY, USA: Association for Computing Machinery, 2019. (IMC '19), p. 116–129. ISBN 9781450369480. Disponível em: <https://doi.org/10.1145/3355369.3355586>.
- [10] FOREMSKI, P.; GASSER, O.; MOURA, G. DNS Observatory: The Big Picture of the DNS. In: *Internet Measurement Conference*. [S.l.: s.n.], 2019. Disponível em: <https://doi.org/10.1145/3355369.3355566>.

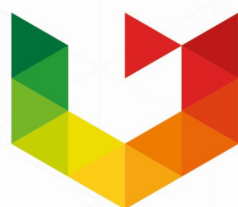
- [11] BÖTTGER, T. et al. An Empirical Study of the Cost of DNS-over-HTTPS. In: *Internet Measurement Conference*. [S.l.: s.n.], 2019. Disponível em: <https://doi.org/10.1145/3355369.3355575>.
- [12] LU, C. et al. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come? In: *Internet Measurement Conference*. [S.l.: s.n.], 2019. Disponível em: <https://doi.org/10.1145/3355369.3355580>.
- [13] de VRIES, W. B. et al. Verfploeter: Broad and load-aware anycast mapping. In: *Proceedings of the ACM Internet Measurement Conference*. London, UK: [s.n.], 2017. Disponível em: <https://ant.isi.edu/datasets/anycast/index.html#verfploeter>.
- [14] ROSSOW, C. Amplification hell: Revisiting network protocols for ddos abuse. In: . [S.l.: s.n.], 2014. ISBN 1-891562-35-5. Disponível em: www.christian-rossow.de/publications/amplification-ndss2014.pdf.
- [15] CALLAHAN, T.; ALLMAN, M.; RABINOVICH, M. On modern dns behavior and properties. *SIGCOMM Comput. Commun. Rev.*, Association for Computing Machinery, New York, NY, USA, v. 43, n. 3, p. 7–15, jul 2013. ISSN 0146-4833. Disponível em: <https://doi.org/10.1145/2500098.2500100>.
- [16] CASTRO, S. et al. Understanding and preparing for DNS evolution. In: *Traffic Monitoring and Analysis, Second International Workshop, TMA 2010, Zurich, Switzerland, April 7, 2010, Proceedings*. [S.l.: s.n.], 2010. p. 1–16. Disponível em: https://doi.org/10.1007/978-3-642-12365-8_1.
- [17] CASTRO, S. et al. A Day at the Root of the Internet. *Computer Communication Review*, v. 38, p. 41–46, 09 2008. Disponível em: <https://doi.org/10.1145/1452335.1452341>.
- [18] WESSELS, D.; FOMENKOV, M. Wow, that's a lot of packets. In: *Proc. of Passive and Active Measurement Workshop*. [S.l.: s.n.], 2003. Disponível em: https://www.caida.org/catalog/papers/2003_dnspackets/wessels-pam2003.pdf.
- [19] SCHOMP, K. et al. On Measuring the Client-Side DNS Infrastructure. In: *ACM Internet Measurement Conference*. [S.l.: s.n.], 2013. p. 77–90. Disponível em: <https://dl.acm.org/doi/pdf/10.1145/2504730.2504734>.
- [20] WICAKSANA, M. A. Ipv4 Vs Ipv6 Anycast Catchment: a Root DNS Study. 2016. Disponível em: http://essay.utwente.nl/70921/1/WICAKSANA_MA_EEMCS.pdf.
- [21] CHOWDHURY, C. *Finding Malicious Usage Via Capture, Storage, Analysis and Visualization of DNS Packets*. [S.l.]: Kansas State University, 2019. Disponível em: <https://krex.k-state.edu/dspace/handle/2097/39482>.

- [22] ROOT-SERVERS. *Root Server Technical Operations Assn.* 2020. Disponível em: <http://www.root-servers.org/>. Acesso: Maio/2020.
- [23] LIU, Z. et al. Two days in the life of the dns anycast root servers. In: *Proceedings of the 8th International Conference on Passive and Active Network Measurement*. Berlin, Heidelberg: Springer-Verlag, 2007. (PAM'07), p. 125–134. ISBN 9783540716167. Disponível em: <https://dl.acm.org/doi/10.5555/1762888.1762906>.
- [24] BELLIS, R.; BLIGH, A.; WIJNGAARDS, W. *DNS Proxy Bypass by Recursive DNS Discovery and LOCAL.ARPA*. [S.l.], 2009. Disponível em: <http://www.ietf.org/internet-drafts/draft-bellis-dns-recursive-discovery-00.txt>. Acesso: Maio/2020.
- [25] POSTEL, J. *Domain Name System Structure and Delegation*. [S.l.], 1994. Disponível em: <http://www.rfc-editor.org/rfc/rfc1591.txt>. Acesso: Maio/2020.
- [26] LIU, B. et al. Who is answering my queries: Understanding and characterizing interception of the dns resolution path. In: . [S.l.: s.n.], 2018. Disponível em: <https://dl.acm.org/doi/10.1145/3340301.3341122>.
- [27] IANA. *Domain Name System (DNS) Parameters*. 2021. Disponível em: <https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml>. Acesso: Março/2021.
- [28] LOTTOR, M. *Domain Administrators Operations Guide*. [S.l.], 1987. Disponível em: <https://tools.ietf.org/html/rfc1033>. Acesso: Maio/2020.
- [29] MOURA, G. C. et al. Cache me if you can: Effects of DNS time-to-live. *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, n. 1, p. 101–115, 2019. Disponível em: <https://doi.org/10.1145/3355369.3355568>.
- [30] ZYL, I. van; RUDMAN, L. L.; IRWIN, B. A review of current DNS TTL practices. *Satnac*, n. September 2015, 2015. Disponível em: https://www.researchgate.net/publication/327622760_A_review_of_current_DNS_TTL_practices.
- [31] SARIDOU, B.; SHIAELES, S.; PAPADOPOULOS, B. DDoS attack mitigation through Root-DNS Server: A case study. In: . [S.l.]: IEEE, 2019. p. 60–65. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8817238>.
- [32] de O. SCHMIDT, R.; HEIDEMANN, J.; KUIPERS, J. Anycast latency: How many sites are enough? University of Southern California, n. ISI-TR-2016-708, 5 2016. Disponível em: <https://research.utwente.nl/en/publications/anycast-latency-how-many-sites-are-enough>. Acesso: Junho/2020.
- [33] MOURA, G. C. et al. Anycast vs. DDoS: Evaluating the November 2015 Root DNS Event. *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*,

- n. November 2015, p. 255–270, 2016. Disponível em: <https://dl.acm.org/doi/10.1145/2987443.2987446>.
- [34] EASTLAKE, D. E.; KAUFMAN, C. Domain name system security extensions. *RFC*, v. 2065, p. 1–41, 1997. Disponível em: <https://tools.ietf.org/html/rfc2535>. Acesso: Abril/2021.
- [35] ATENIESE, G.; MANGARD, S. A New Approach to DNS Security (DNSSEC). In: *Proceedings of the 8th ACM Conference on Computer and Communications Security*. [S.l.]: Association for Computing Machinery, 2001. p. 86–95. Disponível em: <https://dl.acm.org/doi/10.1145/501983.501996>.
- [36] ROSE, S. et al. *Resource Records for the DNS Security Extensions*. [S.l.], 2005. Disponível em: <https://rfc-editor.org/rfc/rfc4034.txt>. Acesso: Maio/2020.
- [37] THEOBALD, O. *Machine Learning for Absolute Beginners: A Plain English Introduction*. Scatterplot Press., 2017. (Machine Learning from Scratch Series). ISBN 9781549617218. Disponível em: <https://books.google.com.br/books?id=PGNzswEACAAJ>.
- [38] SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, v. 3, n. 3, p. 210–229, 1959. Disponível em: <https://ieeexplore.ieee.org/document/5392560>.
- [39] JO, T. *Machine Learning Foundations: Supervised, Unsupervised, and Advanced Learning*. 1st ed. 2021. ed. [S.l.]: Springer, 2021. ISBN 3030658996,9783030658991.
- [40] MASSARON, J. M. L. *Machine Learning for Dummies*. 2. ed. [S.l.]: Wiley, 2021. (For Dummies). ISBN 9781119724056, 1119724058, 9781119724063, 1119724066.
- [41] HILBE, J. M. *Practical guide to logistic regression*. [S.l.]: Taylor Francis, 2016. ISBN 9781498709576; 1498709575.
- [42] REDDY, A. et al. Using gaussian mixture models to detect outliers in seasonal univariate network traffic. *2017 IEEE Security and Privacy Workshops (SPW)*, p. 229–234, 2017. Disponível em: <https://ieeexplore.ieee.org/document/8227312>.
- [43] FLACH, P. *Machine Learning: The Art and Science of Algorithms that Make Sense of Data*. [S.l.]: Cambridge University Press, 2012. ISBN 1107422221, 978-1107422223.
- [44] BAHROLOLUM, M.; KHALEGHI, M. Anomaly intrusion detection system using gaussian mixture model. In: *2008 Third International Conference on Convergence and Hybrid Information Technology*. [S.l.: s.n.], 2008. v. 1, p. 1162–1167. Disponível em: <https://ieeexplore.ieee.org/document/4682192>.
- [45] DNS-OARC. *The DNS Operations, Analysis, and Research Center*. 2020. Disponível em: <https://www.dns-oarc.net/>. Acesso: Abril/2021.

- [46] DITL. *Day In The Life of the Internet*. 2020. Disponível em: <https://www.dns-oarc.net/oarc/data/ditl>. Acesso: Março/2021.
- [47] MOURA, G. C. et al. When the dike breaks: Dissecting DNS defenses during DDos. *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, n. September, p. 8–21, 2018. Disponível em: <https://doi.org/10.1145/3278532.3278534>.
- [48] PANG, J. et al. On the responsiveness of DNS-based network control. *Internet Measurement Conference*, p. 21–26, 2004. Disponível em: <https://dl.acm.org/doi/10.1145/1028788.1028792>.
- [49] DANZIG, P. B.; OBRACZKA, K.; KUMAR, A. An analysis of wide-area name server traffic: A study of the internet domain name system. *SIGCOMM Comput. Commun. Rev.*, Association for Computing Machinery, New York, NY, USA, v. 22, n. 4, p. 281–292, out. 1992. ISSN 0146-4833. Disponível em: <https://doi.org/10.1145/144191.144301>.
- [50] JUNG, J. et al. Dns performance and the effectiveness of caching. In: *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. New York, NY, USA: Association for Computing Machinery, 2001. (IMW '01), p. 153–167. ISBN 1581134355. Disponível em: <https://doi.org/10.1145/505202.505223>.
- [51] LENTZ, M. et al. D-mystifying the d-root address change. In: *Proceedings of the 2013 Conference on Internet Measurement Conference*. New York, NY, USA: Association for Computing Machinery, 2013. (IMC '13), p. 57–62. ISBN 9781450319539. Disponível em: <https://doi.org/10.1145/2504730.2504772>.
- [52] KAI, K. S. B.; CHONG, E.; BALACHANDRAN, V. Anomaly detection on dns traffic using big data and machine learning. 2019. Disponível em: <http://ceur-ws.org/Vol-2622/paper14.pdf>.
- [53] NIC.AT. *Global TLD Report 2019*. 2019. Disponível em: <https://www.nic.at/media/files/Statistiken/CENTR/CENTRstats-Global-TLD-Report-2019-3.pdf>. Acesso: Abril/2021.
- [54] SIDNLABS. *.nl stats and data*. 2020. Disponível em: <https://stats.sidnlabs.nl/en/>. Acesso: Maio/2021.
- [55] WIRESHARK. *Man pages*. 2020. Disponível em: <https://www.wireshark.org/docs/man-pages/tshark.html>. Acesso: Outubro/2020.
- [56] IANA. *Root Zone File*. 2020. Disponível em: <https://www.internic.net/domain/root.zone>. Acesso: Março/2021.
- [57] SPEARMAN, C. The proof and measurement of association between two things. *American Journal of Psychology*, v. 15, p. 88–103, 1904. Disponível em: <https://doi.org/10.2307/1412159>.

- [58] VANDERPLAS, J. *Python Data Science Handbook: Essential Tools for Working with Data*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2016. ISBN 1491912057.
- [59] YILDIRIM, S. *DBSCAN Clustering — Explained*. 2021. Disponível em: <https://towardsdatascience.com/dbscan-clustering-explained-97556a2ad556#:~:text=DBSCAN%20stands%20for%20density%2Dbased,many%20points%20from%20that%20cluster>. Acesso: Abril/2021.
- [60] SCIKIT. *User Guide*. 2021. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>. Acesso: Abril/2021.
- [61] SCIKIT. *Silhouette Coefficient*. 2021. Disponível em: <https://scikit-learn.org/stable/modules/clustering.html#silhouette-coefficient>. Acesso: Maio/2021.
- [62] SCHWARZ, G. Estimating the Dimension of a Model. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 6, n. 2, p. 461 – 464, 1978. Disponível em: <https://doi.org/10.1214/aos/1176344136>.



UPF

UNIVERSIDADE
DE PASSO FUNDO

UPF Campus I - BR 285, São José
Passo Fundo - RS - CEP: 99052-900
(54) 3316 7000 - www.upf.br