

Análise da Funcionalidade da NWDAF no Core 5G Sobre um Conjunto de Dados

Leonardo Azalim de Oliveira¹, Rodrigo Oliveira Silva¹, Pedro Campos Lima¹,
Antônio Marcos Souza Pereira¹, Júlia Almeida Valadares¹,
Edelberto Franco Silva¹, Mário Antônio Ribeiro Dantas¹

¹Universidade Federal de Juiz de Fora – UFJF – Juiz de Fora – MG – Brazil

{leonardo.azalim, rodrigooliveira, pedrolima, antonio.marcos,
juliavaladares, edelberto, mario.dantas}@ice.ufjf.br

Abstract. *The 5G technology is an evolution of the mobile networks due to the service-oriented paradigm that allows greater flexibility in management and the potential application of data analysis. The 3GPP defines the NWDAF as the network function responsible for data analysis in 5G, however, the literature still lacks works on this function. In order to fill this gap, this paper investigates the NWDAF using a dataset from a simulated 5G network and employing machine learning models as network protocol classifiers. The results have an accuracy of approximately 75%, providing not only reproducibility but also paving the way for future investigations in data analysis as specified by the 3GPP.*

Resumo. *A tecnologia 5G representa uma evolução nas redes móveis por conta do paradigma orientado a serviços que permite mais flexibilidade de gerenciamento e a possibilidade da aplicação da análise de dados. O 3GPP define a NWDAF como a função de rede responsável pela análise de dados no 5G, porém, a literatura ainda é escassa de trabalhos sobre esta função. A fim de preencher esta lacuna, este artigo investiga a NWDAF sob um conjunto de dados de uma rede 5G simulada utilizando modelos de aprendizado de máquina como classificadores de protocolos de rede. Os resultados apresentam acurácia em torno de 75%, e além disso, possibilitam a reprodutibilidade e investigações futuras em análise de dados em acordo com o especificado pelo 3GPP.*

1. Introdução

A quinta geração de redes móveis (5G) trouxe uma mudança significativa no contexto de redes móveis com o paradigma da orientação a serviços. Essa nova abordagem permitiu a extensiva virtualização da infraestrutura da rede 5G, aumentando consideravelmente a flexibilidade na configuração e instalação de ambientes de redes móveis baseados nesta geração quando comparada com as gerações anteriores. Por sua vez, o *3rd Generation Partnership Project* (3GPP) é um órgão internacional de colaboração especialmente importante na padronização do cenário de redes de telecomunicações móveis. De forma semelhante ao ocorrido na geração anterior (4G), a arquitetura de rede 5G¹ é definida por meio de documentos públicos chamados de *Releases*. Estes são documentos que detalham os aspectos técnicos para orientar a implementação das tecnologias envolvidas em ambientes de produção.

¹<https://www.3gpp.org/technologies/5g-system-overview>

A arquitetura de 5G é composta por elementos chamados de *Network Functions* (NFs). As NFs são elementos modulares que desempenham tarefas específicas e possuem nomes sugestivos, propiciando bons níveis de flexibilidade e eficiência da rede como um todo. A especificação [3GPP 2023] introduz uma interconexão entre a *Network Data Analytics Function* (NWDAF) e demais NFs com o objetivo de estabelecer uma rede coesa e eficiente. As interfaces e serviços internos de comunicação desempenham um papel essencial nesse contexto, facilitando a troca de informações entre o NWDAF e outros elementos do sistema. Através de *Application Programming Interfaces* (APIs) e serviços integrados, a [3GPP 2023] visa otimizar a colaboração e a sincronização entre as diversas NFs presentes em uma rede 5G.

Somente a partir do especificado no *Release 17* em [3GPP 2022], a NWDAF tem seus componentes *Analytics Logical Function* (AnLF), *Model Training Logical Function* (MTLF) e *Analytics Data Repository Function* (ADRF) definidos em detalhes. Porém, um número reduzido de trabalhos investigam a aplicação desta NF no núcleo do 5G [Kim et al. 2022, Pell et al. 2023], além de haver uma grande carência na avaliação desta NF aplicada à análise de dados reais típicos do ambiente de redes de computadores.

A fim de preencher esta lacuna, este artigo estende o trabalho de [Kim et al. 2022] através da investigação da NWDAF sob um conjunto de dados de uma rede 5G utilizando modelos de aprendizado de máquina (Árvore de Decisão, Perceptron Multicamadas e Floresta Aleatória) como classificadores de protocolos de rede. Em conformidade com o *Release 17* do 3GPP, foram conduzidos experimentos sobre um dos casos de uso especificados pelo documento, o “*UE Communication Information*”. Os resultados são promissores, apresentando acurácia em torno de 75%. Comparado ao estado da arte, este trabalho contribui para a reprodutibilidade e pavimenta o caminho para investigações futuras em análise de dados em acordo com os *Releases* do 5G apresentados pelo 3GPP ao utilizar a função de rede designada para este fim.

São destacadas como principais contribuições deste trabalho:

1. A implementação de código aberto² da funcionalidade de geração de modelos a partir de modelos de *Machine Learning* (ML) para utilização na NF NWDAF;
2. Criação e documentação de um ambiente de testes contendo uma rede 5G simulada para geração de tráfego;
3. Criação de um conjunto de dados de tráfego gerado no ambiente de testes que encontra-se disponibilizado publicamente; e
4. Comparação de três diferentes modelos de ML no contexto de caracterização de tráfego de *User Equipment* (UEs) 5G com posterior avaliação dos resultados.

As demais seções deste trabalho estão divididas da seguinte forma: A Seção 2 apresenta os trabalhos relacionados ao tema estudado. A Seção 3 traz uma visão geral da proposta de implementação da funcionalidade de geração de modelos do NWDAF e apresenta o ambiente de testes construído. Já a Seção 4 detalha os resultados obtidos a partir do experimento com os dados gerados na rede simulada do ambiente e traz discussões em torno dos mesmos. Por fim, a Seção 5 descreve as principais conclusões e apresenta os trabalhos futuros da pesquisa.

²https://github.com/oliveiraleo/mnc_NWDAF

2. Trabalhos Relacionados

Esta seção apresenta o estado da arte relacionado a NWDAF e a arquitetura 5G, destacando as características principais de cada trabalho e possibilidades de pesquisas futuras.

O trabalho de [Pateromichelakis et al. 2019] apresenta uma investigação da integração de análise de dados no sistema de comunicação 5G. São apresentados os requisitos para essa implantação, assim como o estado atual das atividades de padronização. Os autores propõem uma *Service-Based Architecture* (SBA) aprimorada que integra a funcionalidade de análise de dados e permite adaptação de serviços de maneira dinâmica. Com foco na gestão preditiva de recursos de rádio e na utilização de bancos de dados para medições de rádio, este artigo auxilia na visão preliminar ao NWDAF e outros componentes associados. Já, de acordo com [Coronado et al. 2022], a NWDAF é uma peça fundamental para a realização de análises avançadas e gestão proativa em redes 5G e 6G. Esse artigo destaca que a NWDAF é equipada com um mecanismo de ML e se integra à SBA, utilizando o método de publicação/assinatura para recuperar informações das NFs e atualizá-las com os resultados da análise de dados. Essa integração de análises impulsionadas por ML dentro da NWDAF possibilita a detecção de anomalias, o cálculo do desempenho de carga da rede e a previsão de cargas futuras, aprimorando assim a capacidade da rede de se adaptar a condições dinâmicas.

A NWDAF é também investigada por [Duan 2021] de forma a destacar sua função central no suporte ao gerenciamento de serviços. O estudo explora as particularidades da interação da NWDAF com outros módulos do sistema, como por exemplo, outras NFs, *Application Functions* (AFs) ou mesmo funções de *Operations, Administration and Maintenance* (OAM), detalhando como ela coleta dados de múltiplas fontes. Os autores destacam a flexibilidade no caso de implantação da NWDAF como uma *Virtual Network Function* (VNF), o que permite tanto a presença de instâncias únicas quanto múltiplas em um mesmo domínio de rede. O processo de descoberta e seleção da instância adequada da NWDAF para análises específicas é abordado, destacando a função crucial da *Network Repository Function* (NRF) nesse contexto.

Com enfoque em mecanismos robustos de controle de acesso no contexto da 5G, o trabalho de [Pell et al. 2023] apresenta uma abordagem da aplicação de ML supervisionado para a classificação de serviços. A metodologia aplicada pelos autores não só simplifica a identificação de NFs registradas, mas também capacita o aprendizado de regras de fluxo autorizadas para o acesso aos serviços. A principal contribuição deste trabalho reside em sua abordagem para enfrentar as implicações de segurança da arquitetura do *5G Core* (5GC), oferecendo uma solução que utiliza o ML para reforçar o controle de acesso.

Com destaque, o trabalho [Kim et al. 2022] apresenta uma proposta de implementação da NWDAF em forma de *software* de código aberto. A NWDAF desenvolvida foi integrada a um 5GC de redes móveis 5G também de código aberto chamado free5GC. A NWDAF é a NF de 5G que coleta informações de outras NFs como a *Access and Mobility Management Function* (AMF) e a *Session Management Function* (SMF), e as analisa com base em modelos treinados para fornecer informações analíticas na forma estatística ou preditiva. Assim, o trabalho propõe uma implementação da NWDAF no free5GC, que consiste em uma MTLF e uma AnLF. Embora o trabalho tenha verificado a operabilidade da NWDAF e disponibilizado o código, a etapa que envolve o ML focado na área de redes 5G ficou em aberto para futuras pesquisas. Destaca-se que o conjunto

de dados utilizados pelos autores não diz respeito à dados relativos a Redes de Computadores e seus protocolos, mas somente uma prova de conceito da aplicação da predição de utilizando o conjunto de dados MNIST³. Um conjunto de dados com 60.000 imagens em escala de cinza representadas em matrizes 28x28 de 10 dígitos, junto com um conjunto de teste de 10.000 imagens. O conjunto de dados foi utilizado pelos autores com a finalidade da predição dos valores numéricos associados.

A análise dos trabalhos anteriores permitiu destacar os principais pontos a serem melhorados a respeito do tema. Deve-se citar o trabalho [Pateromichelakis et al. 2019], que aponta como pesquisa futura o papel das análises no contexto do *5G System* (5GS), se é uma funcionalidade essencial e o impacto dela caso seja implementada. Então, o trabalho de [Kim et al. 2022] aborda a implementação da NWDAF e é base para a proposta presente neste trabalho e que será detalhada na Seção 3. Ela abordará o aspecto de classificação de tráfego empregando modelos de ML, algo que está diretamente relacionado a área de Redes de Computadores e que, como exposto nesta seção, não havia sido explorada anteriormente.

3. Proposta

A partir da implementação de um protótipo de NWDAF integrado a uma instância de 5GC por [Kim et al. 2022], e do levantamento do estado da arte apresentado na Seção 2 é apresentada esta proposta. Como validação da análise, são apresentados ainda a geração e utilização dos metadados extraídos de pacotes capturados do ambiente de uma instância 5GC, assim como a avaliação por meio de modelos de ML.

Desta forma, esta seção detalha: uma visão geral do cenário de aplicação da solução no 5G (na Subseção 3.1); A arquitetura da solução que é baseada na especificação do 3GPP (na Subseção 3.2); O ambiente de testes utilizado para experimentação (através da Subseção 3.3); As características do conjunto de dados construído neste ambiente (Subseção 3.4); e O experimento de captura de pacotes realizado (na Subseção 3.5).

3.1. Cenário de Uso no 5G

Como cenário de uso da proposta tem-se a aplicação dela a partir da utilização feita por um usuário no ambiente 5G. Desta forma, agrupam-se os componentes do ambiente de avaliação considerando um cenário do tipo *Edge-Fog-Cloud* para classificação de protocolos. Ao se analisar os principais elementos da arquitetura de uma rede 5G e os conceitos de uma arquitetura distribuída do tipo *Edge-Fog-Cloud*, torna-se possível separar esses componentes de forma didática. Assim, o enlace sem fios entre o UE e a *Radio Access Network* (RAN) encontra-se na *Edge*, o 5GC na *Fog* e os demais recursos de rede (como acesso à internet e aos *cloud servers*) na *Cloud*. Além de representar uma organização hierárquica, ou seja, o UE não acessa a internet sem que antes se conecte ao 5GC, esta arquitetura permite associar a localização dos elementos tanto do ponto de vista lógico quanto do ponto de vista físico.

Ao levar em conta um dos casos de uso mais difundidos entre o público em geral que é o de um usuário que utiliza seu dispositivo para navegar na internet, fica mais claro que seu primeiro ponto de contato será com os componentes da *Edge*. Já os componentes

³<https://keras.io/api/datasets/mnist/>

da *Fog* podem estar à alguns saltos de distância que ainda assim é possível que a *Quality of Experience* (QoE) deste usuário não seja comprometida. De forma semelhante, os componentes da *Cloud*, neste exemplo, podem se localizar ainda mais distantes do usuário de forma que os eventuais requisitos de QoE ainda sejam cumpridos sem grande dificuldade.

Em um cenário em que o UE realiza troca de dados sobre serviços e protocolos relacionados ao *ping*, *video* e *web*, é possível, a partir da classificação de tais fluxos de dados, realizar o gerenciamento de recursos relacionados ao 5GC. Deve-se destacar que não é o escopo deste trabalho o escalonamento ou reconfiguração de NFs conforme o fluxo classificado, mas sim servir de base para futuras pesquisas que venham a explorar esses cenários. Como já destacado neste artigo, o cenário de uso escolhido é introduzido no *Release 17* do 3GPP, especificamente o “*UE Communication Information*”.

Após esta introdução ao cenário de uso em questão e a proposta de ambiente para avaliação, é possível detalhar a arquitetura com aplicação de ML no contexto do 5GC e todas as demais características envolvidas na proposta.

3.2. Visão Geral da Arquitetura

O NWDAF é diferente de outras NFs no 5GC por dois motivos principais. O primeiro é que ele exige que os modelos sejam retreinados. Isto é notável pois, ao se considerar uma outra NF, uma vez que esta esteja implantada não é esperado que seu comportamento mude. Entretanto, um modelo de ML é diferente já que está ligado aos dados em que foi treinado. O segundo motivo é que o NWDAF necessita de dados históricos, enquanto outras NFs só precisam conhecer o estado atual da máquina ou sistema.

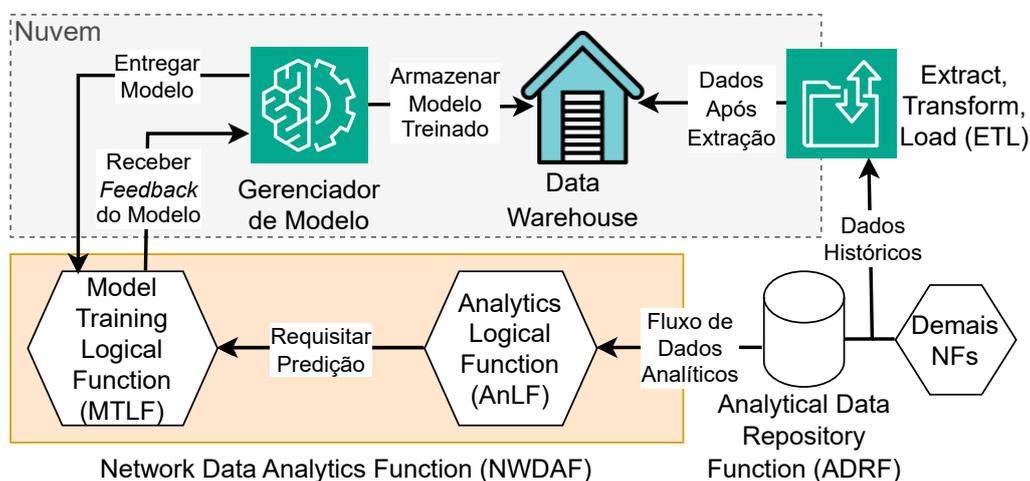


Figura 1. Arquitetura baseada no especificado pela Release 17 [3GPP 2023].

A Figura 1 contém a arquitetura proposta para o *pipeline* de dados e ML sendo esta baseada na especificação [3GPP 2023]. Neste arranjo, os dados históricos devem ser coletados da ADRF e armazenados em um *Data Lake* ou *Data Warehouse*. Dados exigidos pela AnLF para inferência podem vir da ADRF ou de outras NFs. O *pipeline* de dados consiste de uma etapa de *Extract, Transform, Load* (ETL), que possui o propósito de extrair os dados da ADRF e armazená-los em um *Data Warehouse*. Os cientistas de dados então podem treinar seus modelos e, por fim, enviá-los à MTLF, que retorna estatísticas do desempenho deste em um cenário de produção. É importante notar que neste

cenário de produção, tanto as etapas de treinamento do modelo quanto o armazenamento dos dados históricos (representado pelo *Data Warehouse*) podem ser realizados na *Cloud*.

3.3. Descrição do Ambiente de Testes

A partir dos trabalhos [Kim et al. 2022], [Pell et al. 2023] e [de Oliveira and Silva 2023], das especificações do 3GPP cujos principais pontos foram comentados nas seções anteriores, dos conceitos didáticos apresentados na Subseção 3.1 e da visão geral da dinâmica de funcionamento apresentada na Subseção 3.2, um ambiente de testes foi criado.

Por conta da natureza do presente projeto de pesquisa, dos possíveis benefícios da utilização de *software* livre e dos resultados disponíveis em [de Oliveira and Silva 2023] e [Kim et al. 2022], durante a idealização do ambiente de testes foram priorizadas soluções que possuem seu código fonte disponível publicamente. O ambiente criado está representado na Figura 2, que também destaca os nomes das soluções escolhidas para integrá-lo, juntamente com os endereços de *Internet Protocol* (IP) de cada um dos elementos.

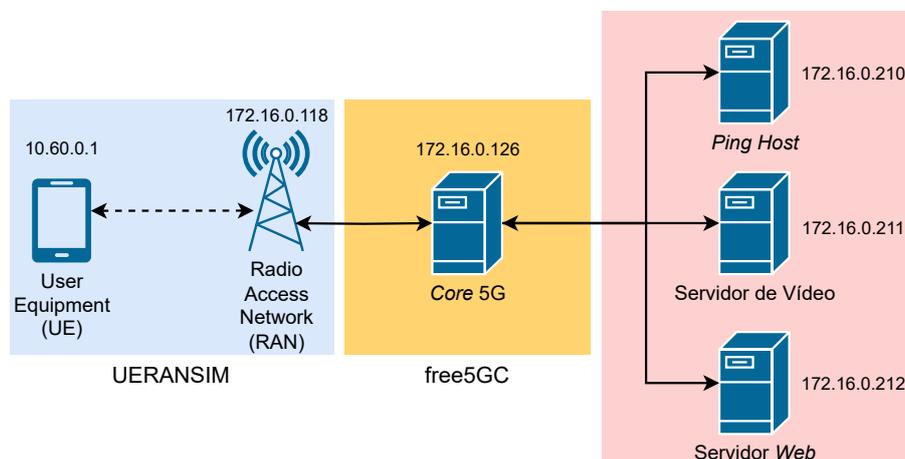


Figura 2. Visão geral do ambiente de testes construído.

O projeto free5GC [free5GC et al. 2020] (destacado na cor amarela), que teve seu código liberado em 2020 e implementa um 5GC, dispõe de um bom nível de documentação⁴ e notável aceitação da comunidade científica já que até o momento da escrita deste artigo, contava com mais de 480 citações no Google Scholar⁵. Os autores do free5GC recomendam⁶ o uso da solução UERANSIM [Güngör et al. 2021] para realizar as simulações da RAN de forma virtual. O UERANSIM (colorido de azul) é uma solução que teve seu código publicado em 2021, pode simular um par de UE e *gNodeB* (gNB) virtuais e também possui aceitação da comunidade contando com mais de 310 citações⁷.

Somando-se à estas soluções já presentes no ambiente, em um terceiro grupo de componentes (que foi destacado pela cor vermelha), foram instanciados 3 servidores: de *ping*, de vídeo e de páginas *web*. Os dois primeiros servidores representam, respectivamente, situações onde a latência e a largura de banda são *Key Performance Indicators* (KPIs) para a QoE do usuário. Por outro lado, quando comparada a estes serviços,

⁴<https://free5gc.org/guide/>

⁵<https://scholar.google.com/scholar?q=free5gc>

⁶<https://free5gc.org/guide/hardware/>

⁷<https://scholar.google.com/scholar?q=UERANSIM>

a navegação *web* não possui requisitos tão restritos, logo, cada um deles representa um contexto de uso distinto.

3.4. Caracterização do Conjunto de Dados

O ambiente de testes apresentado na Subseção 3.3 foi utilizado para criação de um conjunto de dados que foi compartilhado publicamente⁸. No total, foram realizadas seis coletas que foram limitadas a exatamente 1000 pacotes cada. Três coletas foram utilizadas como dados de treinamento e validação e três para os testes de inferência do modelo.

De cada uma das coletas foram extraídas as 7 *features* que se seguem: *No.* (que representa o número sequencial do pacote capturado), *Time* (tempo relativo desde o início da captura até o momento onde aquele determinado pacote foi capturado), *Source* (número de endereço IP de onde partiu o pacote), *Destination* (número de endereço IP do destino do pacote), *Protocol* (*label* com a sigla do protocolo presente no cabeçalho do pacote), *Length* (comprimento total do pacote em *bytes*), e *Info* (outras informações capturadas mas que não pertenciam à nenhuma das demais *features*). A Figura 3 contém uma captura de tela com uma amostra dos dados capturados durante a geração de tráfego via *ping* sendo esta a forma de armazenamento utilizada em todos os demais arquivos *Comma-Separated Values* (CSV) do conjunto de dados.

```
"No.", "Time", "Source", "Destination", "Protocol", "Length", "Info"
"1", "0.000000000", "10.60.0.1", "172.16.0.210", "ICMP", "84", "Echo (ping) request id=0x0009, seq=1/256, ttl=64 (reply in 2)"
"2", "0.001001426", "172.16.0.210", "10.60.0.1", "ICMP", "84", "Echo (ping) reply id=0x0009, seq=1/256, ttl=63 (request in 1)"
"3", "1.002434348", "10.60.0.1", "172.16.0.210", "ICMP", "84", "Echo (ping) request id=0x0009, seq=2/512, ttl=64 (reply in 4)"
"4", "1.003466096", "172.16.0.210", "10.60.0.1", "ICMP", "84", "Echo (ping) reply id=0x0009, seq=2/512, ttl=63 (request in 3)"
"5", "2.004821225", "10.60.0.1", "172.16.0.210", "ICMP", "84", "Echo (ping) request id=0x0009, seq=3/768, ttl=64 (reply in 6)"
"6", "2.005809609", "172.16.0.210", "10.60.0.1", "ICMP", "84", "Echo (ping) reply id=0x0009, seq=3/768, ttl=63 (request in 5)"
"7", "3.006681348", "10.60.0.1", "172.16.0.210", "ICMP", "84", "Echo (ping) request id=0x0009, seq=4/1024, ttl=64 (reply in 8)"
"8", "3.007696589", "172.16.0.210", "10.60.0.1", "ICMP", "84", "Echo (ping) reply id=0x0009, seq=4/1024, ttl=63 (request in 7)"
"9", "4.009171937", "10.60.0.1", "172.16.0.210", "ICMP", "84", "Echo (ping) request id=0x0009, seq=5/1280, ttl=64 (reply in 10)"
"10", "4.010182379", "172.16.0.210", "10.60.0.1", "ICMP", "84", "Echo (ping) reply id=0x0009, seq=5/1280, ttl=63 (request in 9)"
```

Figura 3. Amostra da coleta dos dados de treinamento do servidor de *ping*.

3.5. Descrição do Experimento de Captura

O experimento consistiu em instanciar os elementos do ambiente: servidores, o 5GC e o par de UE e gNB, realizar a conexão entre eles e então utilizar *scripts* e comandos para automatizar a geração de tráfego entre o UE e os três serviços. Durante a execução dos testes, o 5GC além de realizar suas funções normais e o encaminhamento de tráfego, também contava com a execução de uma captura de pacotes que foi realizada pelo utilitário *tshark* na interface de rede criada pelo módulo *GPRS Tunneling Protocol* (GTP) instalado na instância de *free5GC* do ambiente.

Para gerar o tráfego de *ping*, foi utilizado o comando padrão do utilitário *ping* disponível em sistemas operacionais baseados em Linux. O intervalo de envio de pacotes foi, na grande maioria do tempo, mantido em seu valor padrão de 1 segundo, porém também foram enviados alguns pacotes com intervalos de 0,20, 0,25, 0,5, 0,75 e 2 segundos. O tamanho da carga (*payload*) do pacote também foi mantida, na maior parte do tempo, em seu valor padrão (60 *bytes*), entretanto também foram enviados alguns pacotes de 80 e 1400 *bytes* de comprimento. Já para gerar o tráfego de vídeo, foi utilizado o programa *MPV*⁹ que é um *player* de multimídia de código aberto de maneira que o

⁸https://github.com/oliveiraleo/mnc_NWDAF/tree/mnc_Public-5G/ML_test_code/dataset

⁹<https://mpv.io/>

UE reproduziu de forma contínua o arquivo de vídeo armazenado no servidor. Por fim, o tráfego *web* foi criado por meio da execução do utilitário *wget* (com o parâmetro de entrada sendo o arquivo *pagelist.txt*) que era executado de forma contínua pelo UE e então acessava e baixava uma lista predeterminada de páginas do servidor. Após a captura do tráfego, os pacotes foram exportados em formato CSV e então estes novos arquivos foram utilizados como entrada para os modelos de ML que serão abordados na Seção 4.

4. Resultados e Discussões

Com base na proposta apresentada na Seção 3, esta seção destaca os principais resultados obtidos. Primeiramente algumas contribuições da fase inicial do trabalho são explicitadas na Subseção 4.1. Em seguida, a Subseção 4.2 apresenta uma análise mais aprofundada das *features* presentes nos pacotes das coletas. Posteriormente, a Subseção 4.3 explora os modelos de aprendizado de máquina (ML) que foram utilizados, mostra os resultados estatísticos em conjunto com algumas discussões. Por fim, a Subseção 4.4 traz algumas considerações que são pertinentes em relação aos resultados presentes nesta seção.

4.1. Contribuições Iniciais

Durante o processo de validação preliminar do trabalho de [Kim et al. 2022] foram detectadas informações faltantes na documentação do repositório do projeto. Assim, como parte dos resultados deste trabalho, a documentação de suporte do trabalho de [Kim et al. 2022] foi melhorada¹⁰. Também como parte desse processo, o código fonte do módulo NWDAF foi atualizado¹¹ para funcionar corretamente com a versão 3.3.0 do projeto free5GC. A atualização foi verificada por meio de um teste de execução no ambiente construído e o resultado desta verificação pode ser visto na Figura 4 que mostra uma captura de tela exibindo os *logs* da execução do módulo NWDAF.

```
netlab@free5gc-nwdaf:~/free5gc/nwdaf$ ./nwdaf
2023-12-05T13:06:19.858185361Z [INFO][NWDAF] nwdaf
2023-12-05T13:06:19.858249859Z [INFO][NWDAF] NWDAF version:
  Not specify ldflags (which link version) during go build
  go version: go1.18.10 linux/amd64
2023-12-05T13:06:19.858731339Z [INFO][NWDAF][CFG] config version [1.0.1]
2023-12-05T13:06:19.858770515Z [INFO][NWDAF][Init] Server started
2023-12-05T13:06:19.858790546Z [INFO][NWDAF][Util] nwdafconfig Info: Version[1.0.1] Description[NWDAF initial local configuration]
[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env:   export GIN_MODE=release
- using code:  gin.SetMode(gin.ReleaseMode)

[GIN-debug] POST    /nwdaf-mlf/v1/          --> nwdaf.com/mlf.Index (3 handlers)
[GIN-debug] POST    /nwdaf-mlf/v1/:training --> nwdaf.com/mlf.requestModelTraining (3 handlers)
[GIN-debug] POST    /nwdaf-anlf/v1/        --> nwdaf.com/anlf.Index (3 handlers)
[GIN-debug] POST    /nwdaf-anlf/v1/:inference --> nwdaf.com/anlf.requestModelInference (3 handlers)
2023/12/05 13:07:44 #####3
*****map[Content-Length:[122] Content-Type:[text/html; charset=utf-8] Date:[Tue, 05 Dec 2023 13:07:48 GMT] Server:[Werkzeug/3.0.1 Python/3.8.10]]
*****
map[data:training finish nfService:training(reply) reqNFInstanceID:testhi reqTime:2023-12-05 13:07:44]
2023-12-05T13:07:48.940906955Z [INFO][NWDAF][GIN] | 200 | 127.0.0.1 | POST | /nwdaf-mlf/v1/:training |
2023/12/05 13:07:53 map[data:0 nfService:inference reqNFInstanceID:Test NF-Function reqTime:2023-12-05 13:07:53]
map[Content-Length:[121] Content-Type:[text/html; charset=utf-8] Date:[Tue, 05 Dec 2023 13:07:53 GMT] Server:[Werkzeug/3.0.1 Python/3.8.10]]
*****
map[data:7 nfService:inference(reply) reqNFInstanceID:Test NF-Functionhi reqTime:2023-12-05 13:07:53]
2023-12-05T13:07:53.278822232Z [INFO][NWDAF][GIN] | 200 | 127.0.0.1 | POST | /nwdaf-anlf/v1/:inference |
```

Figura 4. Teste de execução do código atualizado.

¹⁰https://github.com/net-ty/mnc_NWDAF/pull/8

¹¹https://github.com/oliveiraleo/mnc_NWDAF/commit/1d5fa3f6c4ef5590233322aa5122b0bba442896f

4.2. Análise Exploratória do Conjunto de Dados

Ao se realizar uma análise preliminar das *features* disponíveis no conjunto de dados (que foram apresentadas na Subseção 3.4), foi constatado que o comprimento dos pacotes capturados possuía o comportamento retratado pela Figura 5. Ela contém a distribuição da frequência de ocorrência em escala logarítmica comparada ao tamanho (em *bytes*) dos pacotes capturados, de forma que cada uma das coletas encontra-se representada por uma cor diferente. É possível notar que, na maior parte do tempo, existe uma relação nos padrões de captura dos dados utilizados para inferência e treinamento ao se considerar um mesmo serviço. Por exemplo, ao se considerar o *ping*, ocorre uma considerável concentração de pacotes de tamanho igual a 84 *bytes*. Por outro lado, tanto o serviço de vídeo quanto o *web*, possuem uma concentração de ocorrência dos comprimentos menores ou iguais a 69 *bytes* e igual a 1400 *bytes*. A principal diferença entre eles é de que enquanto o tráfego de vídeo está em sua grande maioria concentrado em 52 e 1400 *bytes*, o *web* encontra-se mais distribuído em quase todos os valores registrados.

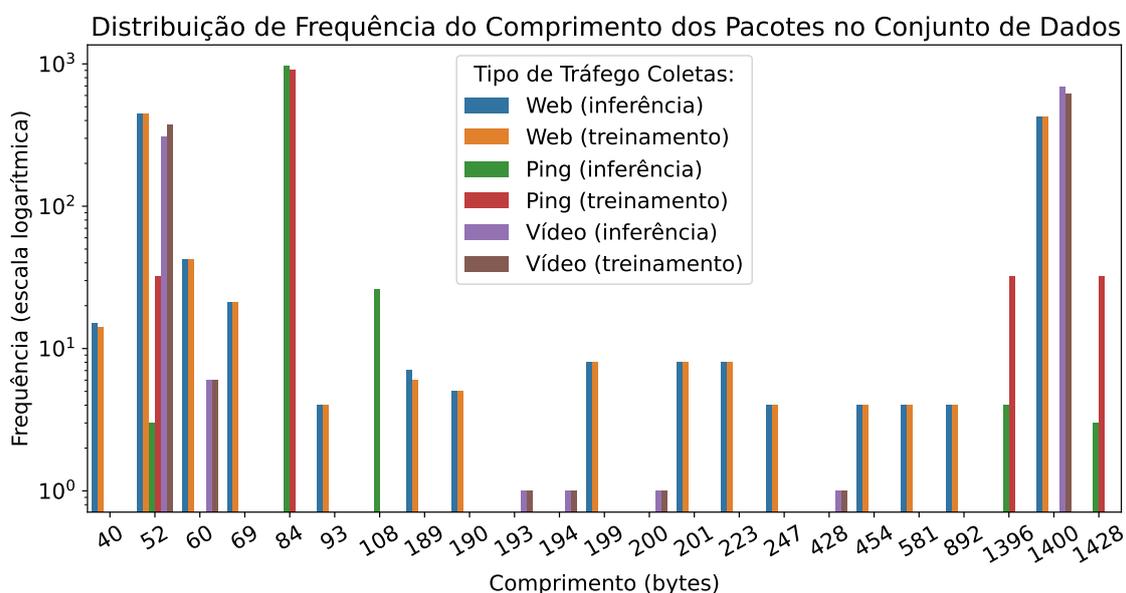


Figura 5. Comprimento dos pacotes do conjunto de dados.

Provavelmente este comportamento foi observado por conta da própria natureza do tráfego dos serviços relacionados. O arquivo de vídeo possui dezenas de *megabytes* de comprimento e é transmitido em forma de um fluxo contínuo de dados. Já a navegação *web* conta com a transmissão de diversos objetos de diferentes tamanhos que são carregados à medida que o *script* navega pelas páginas. Por fim, o tamanho padrão do *ping* do sistema operacional Linux utilizado para instanciar o UE virtual era de 60 *bytes* de carga que são adicionados aos *bytes* do cabeçalho do pacote. Sob outra perspectiva de análise, a Figura 6 apresenta a frequência de ocorrência dos protocolos em cada uma das três coletas realizadas para criar os arquivos de dados de treinamento. Assim, ao se analisar esta *feature* comparando-se os três serviços, é possível constatar que o *ping* se destaca com protocolos bem distintos dos outros dois serviços. Apesar de tanto o vídeo quanto o *web* terem sido instanciados por meio de servidores *Hypertext Transfer Protocol* (HTTP), o protocolo que predominou em ambas as capturas foi o *Transmission Control Protocol* (TCP), seguido do próprio HTTP. Dois motivos principais podem ser identificados

como responsáveis por esses comportamentos: (i) a maneira como os serviços funcionam (ou mais especificamente, qual protocolo utilizam), já que o TCP deve ter sido utilizado para transmissão dos dados em si e o HTTP ficado mais responsável pelo controle e comando das requisições feitas aos servidores; e (ii) a forma com que os serviços de vídeo e *web* foram configurados, por meio da utilização de servidores de similar funcionamento, o que possivelmente contribuiu para influenciar os resultados dos modelos de ML que serão abordados na subseção seguinte.

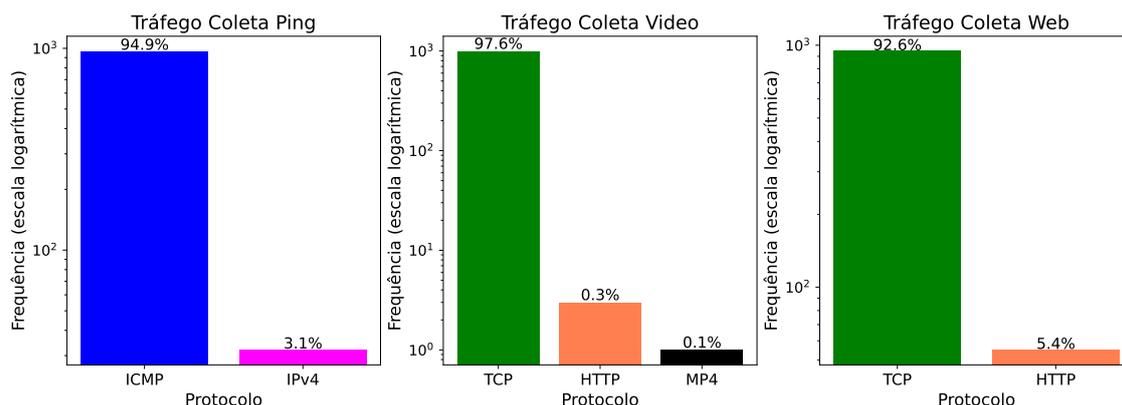


Figura 6. Protocolos dos cabeçalhos dos pacotes do conjunto de dados.

É preciso notar ainda que os endereços IP dos sistemas envolvidos foram mantidos fixos para melhor identificação em uma posterior análise das capturas de pacotes. Entretanto, os IPs foram descartados como *features* do processo de criação do modelo porque, em uma rede real, os endereços podem variar frequentemente, logo, não era desejável neste momento vincular a criação do modelo à eles.

4.3. Utilização de Modelos de Aprendizado de Máquina

Uma área de aprimoramento adicional identificada no NWDAF é a integração e utilização de modelos de ML. Nesse contexto, foram empregados três modelos distintos: Árvore de Decisão [Safavian and Landgrebe 1991], Perceptron Multicamadas [Pal and Mitra 1992] e Floresta Aleatória [Breiman 1996]. Além disso, para avaliar o desempenho desses modelos, foram empregadas métricas que incluem Acurácia, Precisão, F1-score, R2-score, *Mean Squared Error* (MSE), *Root Mean Squared Error* (RMSE) e *Mean Absolute Percentage Error* (MAPE). Todas as métricas foram calculadas através do conjunto de teste, que representa 30% do total da base de dados de treinamento.

Os melhores resultados são apresentados na Tabela 1. Eles foram obtidos após um ajuste fino de parâmetros realizado por meio do método de evolução diferencial [Storn and Price 1997]. A moda de cada um dos parâmetros foi calculada com os dados de treinamento após 10 execuções consecutivas do método. Desta forma, observa-se que tanto a Árvore de Decisão quanto a Floresta Aleatória demonstraram acurácia e precisão superiores a 70%. Já nas métricas seguintes a Floresta Aleatória mostra melhor aproveitamento, chegando a 75,8% na precisão, revocação e F1-score, respectivamente.

Através da Figura 7, que contém uma matriz de confusão, é possível reconhecer que a Floresta Aleatória acertou 100% da classe *ping*, 63,5% da *video* e 62,6% da *web*. Deve-se observar que, por conta das medidas adotadas na implementação para melhorar

Modelo	Acurácia	Precisão	Revocação	F1-score	R2-score	MSE	RMSE	MAPE
Floresta Aleatória	0,758	0,758	0,758	0,758	0,641	0,242	0,492	0,276
Perceptron Multicamadas	0,659	0,655	0,755	0,751	0,521	0,231	0,481	0,251
Árvore de Decisão	0,743	0,740	0,751	0,755	0,639	0,242	0,492	0,276

Tabela 1. Desempenho dos classificadores.

a reprodutibilidade dos resultados, tanto a matriz de confusão quanto os melhores resultados de desempenho tendem a convergir para aqueles presentes nesta seção.

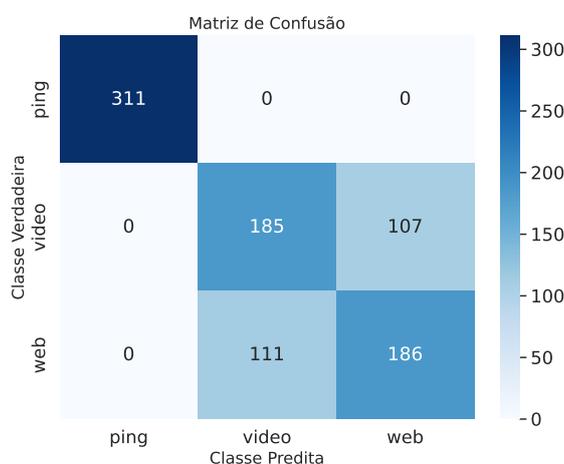


Figura 7. Matriz de confusão das classes do método Floresta Aleatória.

Com o objetivo de identificar quais atributos foram mais relevantes para os modelos, a importância de cada atributo foi extraída. Como pode ser visto na Tabela 2, é notório que o atributo `protocolo` desempenha um papel significativo na identificação de determinadas transferências. Essa distribuição de importância em conjunto com o resultado das análises da Subseção 4.2 pode ajudar a explicar os resultados da matriz da Figura 7, já que foi mais simples diferenciar a classe *ping* das demais através do protocolo utilizado.

Modelo	Protocolo	Tamanho	Duração
Floresta Aleatória	0,622	0,287	0,091
Árvore de Decisão	0,962	0,038	0

Tabela 2. Importância dos atributos para Árvore de Decisão e Floresta Aleatória.

Ainda como parte dos resultados diretamente relacionados ao uso de modelos de ML, foram feitos testes de inferência de classe com coletas presentes no conjunto de dados mas que não foram parte do processo de treinamento. As Tabelas de 3 a 5 apresentam os resultados detalhados deste processo de inferência, onde a classe inferida encontra-se destacada. Os arquivos utilizados foram, respectivamente, capturas de pacotes nos ambientes contendo os serviços *ping*, de vídeo e *web*.

Faz-se necessário destacar alguns detalhes que podem ter influenciado os resultados apresentados. Inicialmente os dados de treinamento foram normalizados e estavam

Arquivo	1ping-capture-intervals-inference			2video-capture-inference			3web-capture-inference		
Classes	ping	video	web	ping	video	web	ping	video	web
Inferência	999	–	–	–	688	311	–	424	575

Tabela 3. Resultados da inferência para o método Floresta Aleatória.

Arquivo	1ping-capture-intervals-inference			2video-capture-inference			3web-capture-inference		
Classes	ping	video	web	ping	video	web	ping	video	web
Inferência	996	3	–	3	684	312	57	424	518

Tabela 4. Resultados da inferência para o método Perceptron Multicamadas.

Arquivo	1ping-capture-intervals-inference			2video-capture-inference			3web-capture-inference		
Classes	ping	video	web	ping	video	web	ping	video	web
Inferência	999	–	–	–	685	314	–	424	575

Tabela 5. Resultados da inferência para o método Árvore de Decisão.

balanceados, com uma coleta de mesmo número de pacotes por classe. Então foi feita a separação entre conjunto de treino e teste (na proporção de 70% e 30%, respectivamente) e a validação não foi realizada por conta do tamanho do conjunto de dados disponível.

4.4. Considerações Finais

Os resultados da Subseção 4.3 foram obtidos excluindo os endereços IP de origem e destino das características analisadas. Em um ambiente como o deste estudo, que é altamente controlado, com tráfego concorrente limitado em relação ao tráfego de interesse, a inclusão desses atributos resulta em todas as métricas dos modelos apresentando uma taxa de acerto de 100%. Esse fenômeno é atribuível ao fato de que cada serviço foi instanciado em um endereço IP específico. A Tabela 6 ilustra que, com a inclusão desses atributos, ambos os modelos avaliados passam a atribuir significativa importância aos endereços IP em comparação com as demais características, uma vez que passam a representar exatamente 49,3% da importância dos modelos Floresta Aleatória e da Árvore de Decisão. Essa descoberta é relevante pois apresenta potencial aplicabilidade em ambientes similares.

Outro ponto a se destacar é de que os três modelos de ML explorados foram escolhidos por conta de suas características. Primeiramente, a Árvore de Decisão tem como pontos fortes a capacidade de lidar tanto com dados categóricos quanto numéricos, e também com conjuntos de dados grandes e complexos, podendo ser mais facilmente visualizadas caso necessário e não requer muito tratamento de dados, sendo resistente à valores ausentes. Então, o Perceptron Multicamadas possui bom desempenho nos processos de treino e inferência, costuma funcionar com boa precisão mesmo quando o conjunto de dados é pequeno e é fácil de aplicar tanto em problemas lineares quanto não lineares. Por fim, a Floresta Aleatória foi escolhida por conta de escalar bem com conjuntos de dados grandes, normalmente atingir bons níveis de precisão, realizar seleção implícita de *features* (já que as mais relevantes para a previsão são automaticamente identificadas com base na frequência e impacto de uso nas decisões de divisão das árvores) e por também possuir boa resistência à *overfitting*.

Modelo	IP Origem	IP Destino	Protocolo	Tamanho	Duração
Floresta Aleatória	0,276	0,217	0,418	0,088	0,003
Árvore de Decisão	0,215	0,278	0,507	0	0

Tabela 6. Importância dos atributos para os modelos adicionando IPs.

Finalmente, levando-se em consideração o contexto e os resultados expostos nesta pesquisa, seria desejável fornecer a funcionalidade de aprendizado em tempo real. De acordo com o especificado pelo 3GPP, isso significaria poder gerar e atualizar o modelo com dados obtidos em tempo real. Vislumbra-se que a melhor forma de realizar essa evolução a partir do estado atual da solução seria por meio de: (i) monitoramento de um ou mais KPIs e por meio de um limiar previamente configurado, acionar a atualização do modelo de forma automática; ou (ii) comandos enviados por um operador/controlador via *Control Plane* (CP). Na primeira opção, devido à quantidade de dados que seriam gerados, possivelmente haveria a necessidade do emprego de tecnologias de armazenamento acopladas a um sistema de *Data Warehouse*.

5. Conclusão

Este artigo apresentou os resultados sobre a análise da funcionalidade NWDAF no 5GC. Utilizando como base o trabalho de [Kim et al. 2022], este trabalho estende suas funcionalidades, avalia sobre um conjunto de dados de redes a classificação com modelos de aprendizado de máquina, e provê documentação e métodos de reprodutibilidade a partir do repositório compartilhado. Os resultados se mostraram promissores, alcançando valores em torno de 75% de acurácia sem a utilização do atributo IP mesmo para modelos simples de análise de dados. A base gerada por este trabalho possibilita a utilização por outros grupos de pesquisa em Redes de Computadores, mais especificamente aqueles com foco em 5G e na função NWDAF para avaliação de métodos de inteligência computacional aplicado no contexto de redes.

No âmbito dos trabalhos futuros, considera-se finalizar a integração do módulo de aprendizado de máquina com o módulo NWDAF que se conecta ao código oficial do free5GC respeitando a especificação [3GPP 2023]. Pretende-se explorar outras técnicas de pré-processamento dos dados e outros modelos de ML com o objetivo de melhorar os resultados de acurácia. Também planeja-se explorar outros cenários e casos de uso dentro daqueles especificados pelo 3GPP [3GPP 2022] bem como expandir as aplicações de serviços presentes no ambiente de testes. De maneira complementar, deseja-se incluir no ambiente de testes um par de UE e gNB reais para se avaliar o comportamento do ambiente em um cenário mais próximo da realidade o que poderia influenciar por exemplo na própria caracterização do tráfego devido à interferências ou adaptações de parâmetros de comunicação de ambos os dispositivos. Finalmente, é almejado o desenvolvimento de um *framework* de experimentação de código aberto que não só possibilite a reprodução dos resultados obtidos mas também sirva de base para futuras validações a cerca do funcionamento do NWDAF em um 5GC.

Agradecimentos

Os autores gostariam de agradecer à Universidade Federal de Juiz de Fora (UFJF) e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio fi-

nanceiro. Os autores também agradecem ao Sr. Frederico de Oliveira Sales pela ajuda na etapa experimental do trabalho e ao Sr. Matheus Brugger Jenevain pelos apontamentos realizados durante a etapa de planejamento do projeto.

Referências

- 3GPP (2022). 3GPP TS 23.288 version 17.4.0 Release 17. Technical report, 3rd Generation Partnership Project.
- 3GPP (2023). 3GPP TS 29.520 version 17.11.0 Release 17. Technical report, 3rd Generation Partnership Project.
- Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.
- Coronado, E., Behraves, R., Subramanya, T., Fernández-Fernández, A., Siddiqui, M. S., Costa-Pérez, X., and Riggio, R. (2022). Zero touch management: A survey of network automation solutions for 5g and 6g networks. *IEEE Communications Surveys & Tutorials*, 24(4):2535–2578.
- de Oliveira, L. A. and Silva, E. F. (2023). Estudo e Avaliação de Métodos de Autenticação EAP na Infraestrutura de Redes de Telecomunicação 5G. In *Anais do Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG)*. SBC.
- Duan, Q. (2021). Intelligent and Autonomous Management in Cloud-Native Future Networks. *Future Internet*, 13(2):42.
- free5GC et al. (2020). free5GC: Open source 5G core network base on 3GPP R15. [Online]. Disponível em: <https://github.com/free5gc/free5gc>.
- Güngör, A. et al. (2021). UERANSIM: Open source 5G UE and RAN (gNodeB) implementation. [Online]. Disponível em: <https://github.com/aligungr/UERANSIM>.
- Kim, T., Kim, J., Ko, H., Seo, S., Jcon, Y., Jeong, H., Lee, S., and Pack, S. (2022). An implementation study of network data analytic function in 5g. In *2022 IEEE International Conference on Consumer Electronics (ICCE)*, page 1–3. IEEE.
- Pal, S. K. and Mitra, S. (1992). Multilayer perceptron, fuzzy sets, classification.
- Pateromichelakis, E., Moggio, F., Mannweiler, C., Arnold, P., Shariat, M., Einhaus, M., Wei, Q., Bulakci, O., and De Domenico, A. (2019). End-to-end data analytics framework for 5g architecture. *IEEE Access*, 7:40295–40312.
- Pell, R., Moschoyiannis, S., and Shojafar, M. (2023). *Machine Learning Driven Secure Access Control for 5g Core Networks*. SSRN.
- Safavian, S. R. and Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674.
- Storn, R. and Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359.