UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Júnia Maísa de Oliveira Pereira

A Framework for 5G Network Data Analytics Function with Emphasis on Anomaly Detection

Belo Horizonte
2024

Júnia Maísa de Oliveira Pereira

A Framework for 5G Network Data Analytics Function with Emphasis on Anomaly Detection

Final Version

Thesis presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: José Marcos Silva Nogueira
Co-Advisor: Daniel Fernandes Macedo

Belo Horizonte
2024

Pereira , Júnia Maísa de Oliveira.

P436f A framework for 5G Network Data Analytics Function with emphasis on anomaly detection [recurso eletrônico] / Júnia Maísa de Oliveira Pereira - 2024.

1 recurso online (74 f. il., color.) : pdf.

Orientador: José Marcos Silva Nogueira.
Coorientador: Daniel Fernandes Macedo.

Dissertação (Mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciências da Computação.
Referências: f. 67-74

1. Computação – Teses. 2. Inteligência artificial – Teses. 3. Tecnologia 5G – redes de computadores - Teses. 4. Detecção de anomalias (computação) - Teses. 5. Aprendizado do computador – Teses. I. Nogueira, José Marcos Silva. II. Macedo, Daniel Fernandes. III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Computação. IV. Título.

CDU 519.6*82(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais - ICEx

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

A Framework for 5G Network Data Analysis Function with Emphasis on Anomaly Detection

JÚNIA MAÍSA DE OLIVEIRA PEREIRA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. José Marcos Silva Nogueira - Orientador
Departamento de Ciência da Computação - UFMG

Documento assinado digitalmente
gov.br
DANIEL FERNANDES MACEDO
Data: 25/06/2024 16:50:28-0300
Verifique em https://validar.iti.gov.br

Prof. Daniel Fernandes Macedo - Coorientado
Departamento de Ciência da Computação - UFMG

Michele Nogueira
Lima:8538020234
9

Signature numérique
de Michele Nogueira
Lima:85380202349
Date : 2024.06.26
10:09:39 -03'00'

Profa. Michele Nogueira Lima
Departamento de Ciência da Computação - UFMG

Digitally signed by Flavio de Oliveira
Silva:77422090634
DN: cn=Flavio de Oliveira
Silva:77422090634, ou=UFU -
Universidade Federal de Uberlandia,
o=ICPEdu, c=BR
Date: 2024.06.26 13:44:35 +01'00'
Adobe Acrobat version: 2020.005.30636

Prof. Flávio de Oliveira Silva
Centro de Ciências Exatas e Tecnologia - UFU

Profa. Tereza Cristina Melo de Brito Carvalho
Escola Politécnica - USP

Documento assinado digitalmente
gov.br
TEREZA CRISTINA MELO DE BRITO CARVALHO
Data: 27/06/2024 12:05:12-0300
Verifique em https://validar.iti.gov.br

Belo Horizonte, 28 de maio de 2024.

Acknowledgments

Being in the Department of Computer Science (DCC) at UFMG and completing my master's degree would be impossible without the contribution of countless people. It is impossible to list them all, but some stand out for their importance:

- My family, for support me in all aspects. Love you;
- Professor Cristiano Maciel, Professor Gustavo Fernandes Rodrigues, Professor Alex Vidigal Bastos and Professor Marco Aurélio Seluque Fregonezi who encouraged me to register and start the master's degree;
- Professor José Marcos and Professor Daniel Macedo, for guiding and teaching me with such patience;
- Wireless Networks Laboratory (Winet) colleagues and professors, for their technical support;
- Data Science and Automatic Verification Laboratory in Udine University (UNIUD), Italy, in special Professor Angelo Montanari, Professor Andrea Brunello and colleagues Nicola Saccomanno and Alberto Marturano;
- The DCC employees, for the great support;
- The CAPES, CNPQ, FUNDEP, and FAPESP, for funding research[1].

Thank you very much!!

[1] Processes 2023/13518-0, 2020/05182-3 and 2018/23097-3.

Resumo

No domínio da tecnologia 5G, a funcionalidade de análise de dados, denominada *Network Dada Analytics Function* (NWDAF), foi introduzida pela primeira vez na versão 16, destacando sua importância, objetivos e requisitos críticos. O NWDAF aproveita a Inteligência Artificial (IA) para melhorar seu desempenho, mas não é um recurso nativo dos núcleos 5G de código aberto, necessitando de sua implementação conforme as necessidades do usuário. No entanto, para utilizadores que desejam integrar o NWDAF ao núcleo, mas não estão familiarizados com a arquitetura principal ou com a integração da Interface de Programação de Aplicações (API), esta tarefa pode representar um desafio significativo, levando potencialmente ao abandono da implementação da análise de dados.

Embora a atual iteração do NWDAF descreva casos de utilização específicos, não fornece alternativas para o desenvolvimento de novos cenários analíticos. Além disso, espera-se que a evolução das aplicações que utilizam 5G amplie os requisitos analíticos, necessitando de algoritmos de análise mais adaptáveis, capazes de acomodar uma variedade de casos de uso de análises do futuros. A literatura apresenta algoritmos de análise para casos de uso específicos, mas não discute a adaptabilidade a novas demandas analíticas ou a facilitação da implementação do NWDAF no núcleo.

Respondendo a esta lacuna, esta dissertação concebe um *framework* para análise de dados de redes 5G projetada para permitir a modificação algorítmica e a incorporação de novos contextos analíticos. O desenho do *framework* considerou as especificações técnicas do 3GPP relacionadas à análise de dados. Sua avaliação foi realizada por meio de um caso de uso centrado na detecção de anomalias em segurança cibernética. Os resultados indicaram que o framework facilita a instalação e a flexibilidade de novos algoritmos e integra-se efetivamente com o núcleo da rede 5G, suprindo as deficiências previamente identificadas.

Palavras-chave: nwdaf; 5G; análise de dados; inteligência artificial; aprendizado de máquina; cibersegurança; detecção de anomalias.

Abstract

In the realm of 5G technology, the data analytics functionality, termed Network Dada Analytics Function (NWDAF), was first introduced in Release 16, highlighting its significance, objectives, and critical requirements. NWDAF leverages Artificial Intelligence (AI) to enhance its performance, but it is not a native feature of open-source 5G cores, necessitating its implementation as user needs dictate. However, for users who wish to integrate NWDAF into the core but lack familiarity with core architecture or Application Programming Interface (API) integration, this task can pose a significant challenge, potentially leading to the abandonment of data analytics implementation. While the current iteration of NWDAF outlines specific use cases, it does not provide alternatives for developing new analytical scenarios. Furthermore, the evolution of 5G-utilizing applications is expected to extend analytical requirements, necessitating more adaptable analysis algorithms capable of accommodating a variety of future use cases. The literature presents analysis algorithms for specific use cases but fails to discuss adaptability to new analytical demands or the facilitation of NWDAF implementation within the core.

Addressing this gap, this dissertation conceives a framework for 5G network data analytics designed to allow algorithmic modification and the incorporation of new analytical contexts. The framework's design considered the 3GPP technical specifications related to data analytics. Its evaluation was conducted through a use case centered on anomaly detection in cybersecurity. The results indicated that the framework eases installation and new algorithm flexibility and integrates effectively with the 5G network core, fulfilling the previously identified deficiencies.

Keywords: nwdaf; 5G; data analytics; artificial intelligence; machine learning; cybersecurity; anomaly detection.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

It is expected that 3.5 billion users will subscribe to fifth-generation (5G) mobile networks globally by 2026 [22]. Average data usage is estimated to be 35 GB/month/user, resulting from 400 use cases of 5G network technology in 70 industry sectors. The use of technologies such as Artificial intelligence (AI), big data, cloud computing, Software Defined Networks (SDN), and Network Function Virtualization (NFV) has been increasing in the industry in recent years [38]. Data quality, quantity, diversity, and privacy are crucial components of data-driven AI applications, each presenting its own unique set of challenges [10]. As mobile telephony advances to 5G network technology, a critical need for advanced prescriptive analytics will arise to enable automated operations for the 5G network and future generations. Intelligent planning, analytics, fault diagnosis, and adaptive optimization capabilities in 5G networks require the continuous introduction of AI as the development of intelligent networks and transform cloud-based resources [74]. For example, AI can assist in managing complex networks to solve system optimization problems and improve user experience.

The analytics of network data can yield significant insights into the functioning and operation of the network. Moreover, this information can be crucial for developing improvements and automated actions within the network to enhance service delivery to the end user. In 5G, data analytics was first introduced by 3rd Generation Partnership Project (3GPP) in Release 16, outlining its importance, goals, and primary requirements. This function was named Network Data Analytics Function (NWDAF). The Technical Specification in its Release 18 provides further details on the NWDAF, such as data input, functions that supply the data, information obtained from the analytics, types of analyses, analytics output, procedures, reference architecture, and the use of AI, among other important details for those wishing to develop NWDAF for their 5G core (5GC). Furthermore, NWDAF is not a native function of open-source 5GCs, necessitating its implementation should the user require it. However, for users who wish to implement NWDAF in the core but are unfamiliar with the core architecture and Application Programming Interface (API) integration, the NWDAF implementation can be challenging. Therefore, users may give up on implementing data analytics. In addition, the current version of NWDAF presents use cases but does not describe alternatives for the need for new analytics use cases. Moreover, the evolution of applications using 5G can increase the analytics use cases, which will require some form of algorithmic flexibility to adapt to different analytics use cases in the future.

In the literature, data analytics in 5G networks approaches the application of Machine Learning (ML) algorithms unsupervised [48] and supervised [7] [8] [12] [60] [63]), federated learning, deep learning [7] [12] [13] [14] [30] [35] [40] [60] [62], and meta-learning [17] for NWDAF contexts, whether centralized [14] [17] [18] [23] [35] [40] [41] [47] [48] [60] [62] [63] or distributed [8] [12] [35] [39] [57], in simulated environments. Few studies propose a computational system for operating an NWDAF for a 5G network core [7] [8] [12] [17] [18] [21] [23] [30] [41] [43] [48] [57] [75]. The literature presents approaches that facilitate the implementation of an NWDAF in the 5GC for specific use cases, disregarding that use cases need to be modified as applications using 5G evolve. Additionally, the literature presents data analytics in 5G through approaches that apply AI to analytics use cases but do not detail how these algorithms will be incorporated into the 5GC. None of the works found offer ease of NWDAF implementation in 5G, nor do they consider the future need for changes in analytics use cases in 5G or future networks.

Given the importance of the information that can be obtained from the data, the need for changes in analytics use cases, and the challenges that may be encountered in implementing NWDAF in the 5GC, this master's thesis presents a framework solution that addresses these topics and evaluates it using an analytics use case. Additionally, the framework is incorporated into the 5GC.

1.1 Problem Description

The NWDAF by the 3GPP is not installed as a native function in the 5GC, and its implementation requires knowledge of the 5G network and Application Programming Interface (API) that can be difficult if the user does not know about. Then, the user can give up installing NWDAF in its 5GC and obtaining necessary information from the network.

Furthermore, data analytics use cases are presented by the 3GPP, but TS did not describe alternatives for approaching new use cases. For example, 5G applications evolution can increase the use case data analytics, which will require flexibility in NWDAF. i.e., many analytics use cases exist, and news use cases may appear in the future.

Hence, the master thesis aims to answer the question: "**Is it possible to design a flexible and modular solution that allows data analytics algorithms replacement/inclusion/exclusion in 5G network cores by the 3GPP‘s NWDAF?**."

This issue requires in-depth analytics and the development of technical and practical solutions. Therefore, it constitutes a legitimate problem for a master's thesis in Computer Science and 5G networks.

1.2 Motivation

My motivation for this master's thesis stems from limitations identified in the scientific literature about data analytics for 5G network technology and my enthusiasm for acquiring knowledge in 5G networks. Then, the difficulty and complexibility of implementing the 5G NWDAF, the necessity to enhance 5G data analytics through computational systems, and the need for flexible systems for 5G data analytics in the future data analytics are the identified gaps in the literature. Hence, these gaps highlight the necessity for research and development in computer science, specifically in 5G network data analytics and 3GPP NWDAF.

.

1.3 Objective

The master's thesis aims to design a 5G network data analytics framework by 3GPP NWDAF analytics TS. The framework should be easy to incorporate into the 5GC, flexible for new algorithms, and easy to update and correct for continuous operation. Furthermore, a computational 5G data analytics framework aims to collect, storage, process, and analyze data, integrate algorithms, provide information about the 5G network through an automated data flow, and be flexible for algorithm changes. Also, the objectives defined are fundamental for the progress of data analytics in 5G networks and personal evolution knowledge in Computer Science.

Table 1.1: Inclusion and exclusion criteria for articles.

Inclusion	**non-inclusion**
Papers that focus in 5G network data analytics and core functions of 5G networks	Papers that was not concentrated in 5G network functions or 5G network core
Papers published in magazines or conferences	Papers not published in magazines or conferences
Papers published in English or Portuguese	Works in other languages
Papers from the year 2020	Papers leading up to 2020

1.4 Methodology

The master thesis conceives a framework for analyzing 5G network data as below.

1.4.1 Literature Review

The review considered research from various sources, including Google Scholar, IEEE, and Elsevier. Relevant keywords such as “NWDAF”, “data analytics”, “5G”, “machine learning”, and “3GPP” were used to search. To ensure the relevance of the selected studies, papers older than four years and other wireless communication technologies unrelated to 5G network technology were excluded. Table 1.1 presents the inclusion/exclusion criteria for the articles. The systematic review analyzed relevant studies to comprehensively understand 5G network data analytics. The presented works are below.

This systematic review presents relevant, up-to-date studies to help readers comprehensively understand 5G network data analytics. In addition, it enables synthesizing existing works on 5G network data analytics. The systematic review is presented in a dedicated chapter.

1.4.2 ML Algorithms Studies

The Machine Learning algorithms studies are crucial to obtain the knowledge before applying it in a specific situation. We present a Section about the evolution of knowledge in ML. Then, a few unsupervised and supervised ML algorithms are evaluated and compared. At this stage, the data set does not originate from the 5GC, nor has it been implemented in the proposed framework. We apply data selection and dimensionality reduction techniques to improve the evaluation algorithms. The studies aim to compare techniques and knowledge about ML algorithms usage.

1.4.3 Data Analytics Framework Conception

We conceived a data analytics solution according to our motivation and objective. The conceived includes architecture design, component selection, and implementation. The steps are detailed below.

- **Requirement specification:** requirements are essential information and decisions for achieving an objective and a priority list. User needs and functionalities are determined.
- **Architecture design:** an architecture with its components is established in 3GPP accordance. I elaborate a diagram with their components and relationships to simplify the understanding of the proposed solution.
- **Implementation:** The implementation integrates all components according to the requirements specification and architecture design decisions. Studies and requirements contribute to the tool selection criteria.

1.4.4 Evaluation

The evaluation chapter presentsML evaluation, solution evaluation, and implementation in the 5GC. TheML algorithm evaluation presents theML results, which justify the algorithms we are using in the framework evaluation. The solution evaluation aims to analyze, test the functionalities, and verify whether the proposed performs the functionalities defined in its conception. We define and present an analytics use case for the evaluation.

- **ML Algorithms Evaluation:** We present theML algorithm evaluations and compare results to apply the bestML algorithm in framework evaluation. We use the F1-score and AUC as the evaluation metrics. The results of theML algorithms studies (Section 1.4.2) present comparative techniques results. We used the better evaluation algorithms in an analytics use case to evaluate our proposal.

- **Framework Evaluation:** The evaluation aims to analyze and test the functionalities and verify if the proposal meets the established requirements in its conception according to the established use case. Evaluation criteria and an analytics use case are determined. This type of evaluation focuses on verifying the functionalities, specifications, user needs, and resources.

- **Integration in a 5GC:** I deploy a 5GC and user equipment (UE) with the framework. Then, we integrate the proposal with a 5GC and present a discussion. The integration in a 5GC checks if our solution is easily integrated into a 5GC. By presenting the integration process, this thesis aims to contribute insights and guidelines for future research and development endeavors in the 5G data analytics.

1.5 Contributions

The work contributions are (i) the conception and evaluation of a framework for analyzing 5G network data that allows the analytics algorithms inclusion to change analytics use cases; (ii) the methods and techniques development for exploring 5G network data using AI; (iii) the presentation of design definitions for a computational system that supports the operation and modification of ML algorithms for 5G network data analytics; (iv) the identification of open-source software to implement the framework able to include algorithms of network data analytics; (v) in-depth study on 5G network data analytics solutions; (vi) ML algorithms comparative studies; and (vii) A anomaly detection analytics. This work contributes to using 5G network data to obtain information that can improve network efficiency.

1.6 Publications

We published three papers on this master's thesis. The first paper is "*Machine Learning Algorithms Applied to Telemetry Data of SCD-2 Brazilian Satellite*", published in Latin America Networking Conference (2022) [69].

The second paper, "*Comparative analytics of Unsupervised Machine Learning Algorithms for Anomaly Detection in Network Data*", published in 2023 IEEE Latin-American Conference on Communications, LATINCOM (2023) [53]. The ML in this paper provided knowledge for the choices and manipulations of algorithms in this master thesis.

The third paper "*Um* framework *NWDAF para algoritmos de análise de dados de rede 5G e além*", published in 2024 IV Workshop de Redes 6G (42° Simpósio Brasileiro de Redes de Computadores E Sistemas Distribuídos - SBRC). Link not yet available until this master's thesis presentation.

Other articles were published during the master's degree and are presented below.

Our paper "*Data Consumption and User Experience in Video Lecture Sessions via Mobile Telephony Network*", published in 2023 IEEE Latin-American Conference on Communications (LATINCOM), Panama City, Panama [54].

Other paper is "*Improved Video QoE in Wireless Networks Using Deep Reinforcement Learning*", published in 2023 19th International Conference on Network and Service Management (CNSM), Niagara Falls, ON, Canada [49].

The paper "*PIPA: Uma solução integradora de políticas de controle de acesso a recursos e de gerenciamento de identidades*", published in Salão de Ferramentas - Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), Brasília, Distrito Federal, Brasil [51].

The paper "*Abordagem confiança zero aplicada a ambientes computacionais big data: um estudo de caso*", published in XXVII Workshop de Gerência e Operação de Redes e Serviços (2022), Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Fortaleza, Ceará, Brasil [52].

1.7 Master Thesis Structure

The organization of the master's thesis is as follows: Chapter 1 introduces the work and presents the motivation, problem, objective, contributions, publications, and thesis structure; Chapter 2 presents fundamental concepts to understand the work; Chapter 3, presents a literature review obtained from the systematic review method; Chapter 4 presents the development of the proposed framework; and Chapter 5 presents a framework evaluation; Chapter 6 concludes the Master Thesis.

Chapter 2

Background

This chapter presents the fundamental concepts to understand this master thesis. Section 2.1 describes terms from the ML used in this work. Section 2.2 introduces the organizations responsible for standardizing mobile telephony and software architecture. Section 2.3 presents relevant information about the NWDAF and AI management in TS. Section 2.4 defines anomaly detection. Finally, Section 2.5 concludes the chapter.

2.1 Machine Learning in Network Data Analytics

AI is an area of computer science with advanced features of modern computing. It can analyze cellular network data. AI is a research area that encompasses the development of intelligent agents capable of learning and adapting to their environment. ML is a subfield of AI that focuses on developing algorithms that can learn from data.

ML for cellular network data analytics can function in various contexts according to the needs of the mobile operator or user equipment utilizing the network (autonomous vehicles, gaming devices, TV video streaming, smartphones, and smartwatches, among others). The necessary steps for data analytics are as follows [64].

- Collect and store the cellular network data;
- Preprocessed removes errors and inconsistencies in data;
- ML model trains with data;
- Predictions or decisions with ML model.

ML models can perform various cellular network data analytics tasks, including anomaly detection, performance prediction, and policy recommendation.

We can apply ML methods (supervised, unsupervised, and semi-supervised) to cellular network data analytics. Supervised learning requires labeled data with correct outcomes. Tasks such as classification and regression use a Supervised learning model. Unsupervised learning handles unlabeled data. Clustering and dimensionality reduction typically use Unsupervised learning models. Semi-supervised learning combines supervised and unsupervised learning. People commonly use semi-supervised learning models for tasks where labeled data is scarce. When applying a ML method, it is essential to verify the efficiency of its outcome and whether the model performs as expected [73]. The most well-known evaluation metrics in supervised and semi-supervised learning are precision, recall, F1-score, ROC curve, and AUC [56].

The precision is the ratio in the equation 2.1, where tp is the number of true positives and fp is the number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative. The best value is one, and the worst value is 0.

$$Precision = \frac{tp}{tp + fp} \tag{2.1}$$

The recall is the ratio in equation 2.2, where tp is the number of true positives and fn is the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples. The best value is one, and the worst value is 0.

$$Recall = \frac{tp}{tp + fn} \tag{2.2}$$

The F1-Score is a harmonic mean between precision and recall, balancing these metrics. Thus, we can compute the using the following formula:

$$F1\_Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.3}$$

The ROC (Receiver Operating Characteristic) curve is a graphical tool that represents the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) for different threshold values. TPR is the proportion of positive values correctly identified by the model. FPR is the proportion of values classified as positive but are negative.

$$FPR = \frac{fp}{fp + tn} \tag{2.4}$$

$$TPR = Recall = \frac{tp}{tp + fn} \tag{2.5}$$

Another commonly used metric is the area Under the ROC Curve (AUC-ROC), which provides a single measure of the overall model performance. The formula for calculating AUC-ROC involves integrating the ROC Curve over its entire range. AUC-ROC ranges from 0 to 1, where 1 represents perfect performance, and 0.5 represents performance equivalent to chance.

2.2 Telephony Cellular Standardization

The 3GPP is an association that brings together a series of telecommunications regulations. The 3GPP is a set of telecommunications associations from the United States, Europe, Japan, South Korea, and China. The 3GPP recognizes mobile networks worldwide patents and TS. In this way, it is possible to ensure that equipment and network device manufacturers develop compatible products. To achieve this, the 3GPP periodically presents relevant information that determines the standard to be followed by cellular mobile telephony operators.

2.2.1 Fifth Generation Network Service-Based Architecture

The fifth generation of mobile networks, defined by the 3GPP [5], a telecommunications technology organization, has introduced two new concepts: 5G New Radio (NR) and Service-Based Architecture (SBA). SBA is a cloud-native service framework supporting the core functionalities of 5G networks through interconnected Network Functions (NFs) on a shared service infrastructure or bus. The NF enables the flexibility and adaptability of 5G networks. They can be deployed and adapted as necessary to fulfill the precise requirements of the network and the services provided to users. Network function virtualization, in particular, is a crucial feature of 5G that enables more efficient and scalable deployment of network infrastructure. An NF provides services accessible to any other authorized NF through Application Programming Interfaces (APIs) named Service-Based Interfaces (SBI). API specifications describe services exposed by one NF (service producer) to another NF (service consumer). These identify the accessible service dataset and indicate the authorized operations on that service data. With 5G SBA, it is possible to utilize NFs from different vendors in the network core [23]. Through interfaces, any NF offers its services to all other authorized NFs or to any "consumers" who can use these provided services [66]. SBA's benefits are modularity, scalability, reliability, cost-effective operation, easy deployments, and faster innovation. Figure 2.1 presents 5G architecture and the network functions.

- **Application Function (AF):** is responsible for policy management and ensuring Quality of Service (QoS) for specific or third-party applications and services. Additionally, the AF monitors network performance regarding defined policies and can make adaptive resource allocation decisions to meet application requirements.

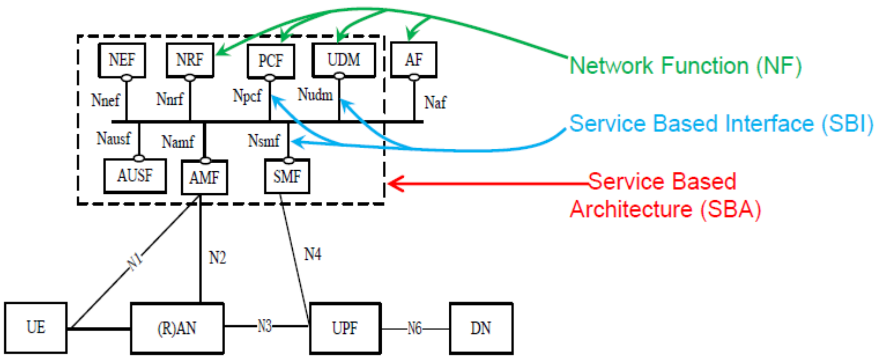


Figure 2.1: Service-Based Architecture (SBA) [65].

- **User Plane Function (UPF):** user plane packet forwarding and routing. Anchor point for mobility;

- **User Equipment (UE):** user equipment are devices that enable access to the mobile cellular telephone network, such as smartphones, tablets, and smartwatches, among others;

- **Radio Access Network (RAN):** system that connects individual devices to other network parts through radio connections.

- **Data Network (DN):** identifies Service Provider services, Internet access, or third-party services.

- **Functions of Service Based Architecture:**

  - **Network Exposure Function (NEF):** responsible for managing external data on the open network. All external applications that want to access the internal data of the 5GC must use NEF;

  - **Network Repository Function (NRF):** stores the characteristics that describe each registered NF and allow other NFs to consult its database to obtain the network address of the desired services;

  - **Policy Control Function (PCF):** unified policy framework to govern network behavior. Provides policy rules for the control plane;

  - **Unified Data Management (UDM):** manages user information. Generates AKA authentication credentials and authorizes access based on subscription data;

  - **Authentication Server Function (AUSF):** authentication for 3GPP access and non-3GPP untrusted access;

  - **Access and Mobility Management Function (AMF):** registration, access control and mobility management;

  - **Session Management Function (SMF):** creates, updates, and removes PDU sessions. SMF manages session context through UPF. UE IP address allocation and Dynamic Host Configuration Protocol (DHCP) function;

In addition, the 5G SBA has Operation, Administration, and Maintenance (OAM). The OAM, in the context of 5G 3GPP, refers to a set of resources and protocols used to manage and maintain active 5G communication networks. To achieve this, OAM performs various actions such as performance monitoring, fault management, software updates, and security management.

2.3 NWDAF 3GPP Technical Specifications

The 3GPP TS present extensive NWDAF support procedures. The section presents TS related to the NWDAF, NWDAF security, and AI in the Release 18. As a result, we study three related TS: TS 29.520, TS 23.288 and TS 33.521. TS 29.520 V18.3.0 [6] presents network data analytics service specifications, including APIs. TS 23.288 V18.3.0 [3] introduces the network analytics function, desired operation, and architecture. TS 33.521 [2] presents specific requirements and test cases for NWDAF.

2.3.1 5G Network Data Analytics Function

The NWDAF provides NF service consumers information on different analytics events. The NWDAF allows NF service consumers to subscribe to and unsubscribe from one-time, periodic notifications or notifications when an event is detected. NWDAF allows NF service consumers to request the transfer of subscriptions for analytics events [6]. Analytics information is either statistical information of past events or predictive information [3]. Figure 2.2 presents an NWDAF in 5GC SBA.

The NWDAF includes one or more of the following functionalities [4]:

- Support data collection from NFs and AFs;
- Support data collection from OAM;
- Support retrieval of information from data repositories (e.g., UDR via UDM for subscriber-related information or via NEF(PFDF) for PFD information);

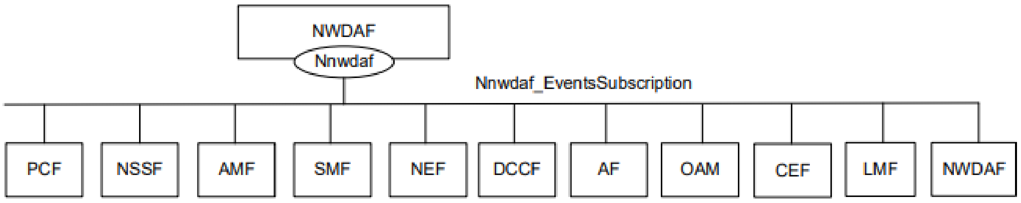


Figure 2.2: Reference architecture for the Nnwdaf_EventsSubscription service; SBI representation TS 29.520 [6].

- Support data collection of location information from LCS system;
- NWDAF service registration and metadata exposure to NFs and AFs;
- Support analytics information provisioning to NFs and AFs;
- Support ML model training and provisioning to NWDAFs (containing Analytics logical function);
- Support bulked data related to Analytics ID(s) provisioning for NFs;
- Support accuracy information about Analytics IDs provisioning for NFs;
- Support accuracy information or accuracy degradation about ML model provisioning for NFs;
- Support roaming exchange capability to exchange data and analytics between PLMNs;
- Support Federated Learning (FL) to train an ML model among multiple NWDAF (containing MTLF).

The NWDAF interacts with different entities for different purposes [3]:

- Data collection based on subscription to events provided by AMF, SMF, UPF, PCF, UDM, NSACF, AF (directly or via NEF) and OAM;
- (Optionally) Analytics and Data collection using the DCCF (Data Collection Coordination Function);
- Retrieval of information from data repositories (e.g., UDR via UDM for subscriber-related information or via NEF(PFDF) for PFD information);
- Data collection of location information from Location Based Services (LCS) system;
- (Optionally) Storage and retrieval of information from ADRF (Analytics Data Repository Function);

- (Optionally) Analytics and Data collection from MFAF (Messaging Framework Adaptor Function);
- Retrieval of information about NFs (e.g., from NRF for NF-related information);
- On-demand provision of analytics to consumers, as specified in clause 6, TS 23.288.
- Provision of bulked data related to Analytics ID(s);
- Provision of Accuracy information about Analytics ID(s);
- Provision of ML model accuracy information or ML model accuracy degradation about an ML Model.

The functionalities are related to the type of events observed. The types of observed events include [6]: slice load level information, network slice instance load level information, service experience, NF load, network performance, abnormal behavior, UE mobility, UE communication, User data congestion, QoS sustainability, Dispersion, Redundant transmission experience, SM congestion control experience, WLAN performance, DN performance, PFD determination, PDU session traffic, movement behavior, and location accuracy.

2.3.1.1 The NWDAF Architecture

The NWDAF is responsible for the analytics and exposition of data within the core of the 5G network. For that, it presents two requirements: data collection and analytics exposition. Figure 2.3 presents the NWDAF architecture outlined in 3GPP Release 15 [29]. The NF/OAM/AF, acting as the Analytical Consumer, requests analyses from the NWDAF.

The NWDAF receives updates with each TS published by the 3GPP. It contains five logical functions introduced in Rel-17, also referred to as components: the Analytical Logical Function (AnLF), the Model Training Logical Function (MTLF), the Data Collection Coordination (and Delivery) Function (DCCF), the Analytical Data Repository Function (ADRF), and the Message Framework Adapter Function (MFAF).

The Analytical Logical Function (AnLF) collects the analytical request and sends the response to the consumer. The AnLF requires historical data that the model needs for prediction. It requests historical data from the DCCF (Data Collection Coordination and Delivery Function).

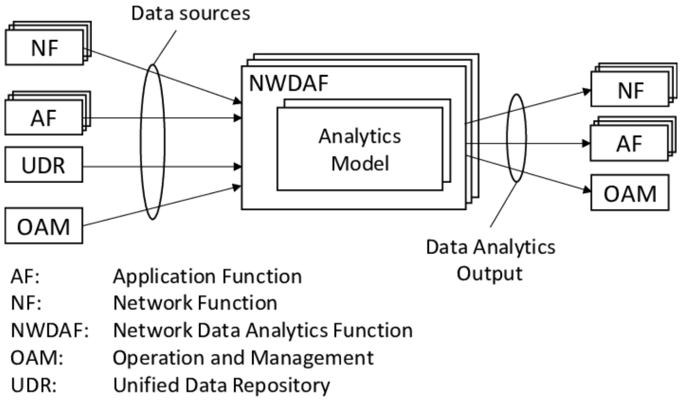


Figure 2.3: NWDAF architecture 3GPP Rel-15 [29].

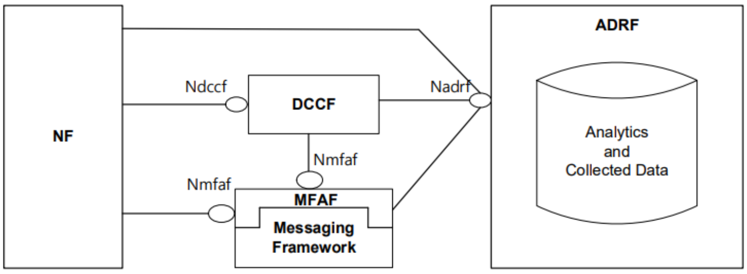


Figure 2.4: Data storage architecture for analytics and collected Data [3].

Model Training Logical Function (MTLF) is responsible for training and deploying the model inference microservice. Accepted data formats include ML code for online training, code of saved models with their parameters and already-trained or pre-trained models, and container images. The MTLF produces ML algorithms used by the Analytical Logical Function (AnLF) to provide the insights that will empower the 5G network at various deployment locations. NWDAF can contain an MTLF, an AnLF, or both logical functions. In addition, ADRF, DCCF, and MFAF components are described below

The 5G System architecture allows Analytics Data Repository Function (ADRF) to store and retrieve the collected data and analytics, Figures 2.4. One or more ADRF instances can be deployed in the network to store raw data or associated analytics that have been performed on that data. The NWDAF, DCCF, or MFAF can store, access, and delete data and analytics as required. In addition, the ADRF can be instructed to subscribe to event notifications to allow data or analytics to be automatically harvested by the ADRF.

The Data Collection Coordination (and Delivery) Function (DCCF) is responsible for monitoring actively collected data from the sources it coordinates, Figure 2.5. It serves as the central point for managing all data requests. If another NF has already requested the same dataset and these data are available, DCCF sends them directly to NWDAF. Otherwise, DCCF initiates a data transfer from the Data Provider. It also initiates data

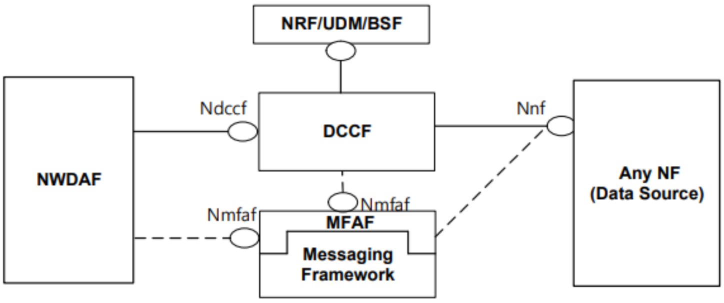


Figure 2.5: Data Collection architecture using Data Collection Coordination [3].

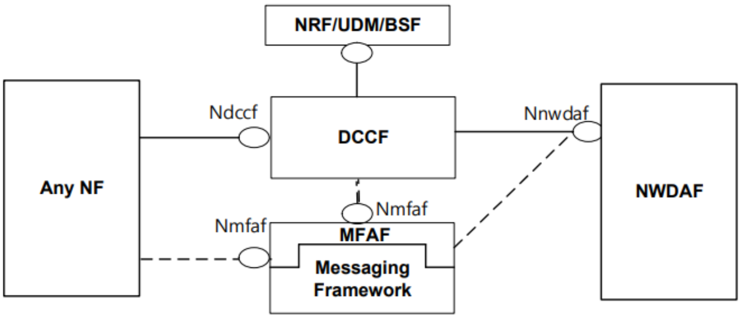


Figure 2.6: Network Data Analytics Exposure architecture using Data Collection Coordination [3].

transfer with the data provider. DCCF forwards the data to AnLF.

The Messaging Framework Adapter Function (MFAF) provides a messaging framework to receive data from DCCF, process, format, and send data to consumers or notification terminals, Figure 2.6. Data transfer will occur between MFAF and ADRF. It is important to note that the ML model's performance degrades over time, which can negatively impact system performance, such as excessive resource allocation or experience. Therefore, continuous self-monitoring and retraining of models are necessary. Retraining with more recent data can be managed by MTLP or outside the edge/core. In some instances, a new ML model design may be necessary. When retraining is no longer effective, in MTLP, MTLP needs to send an alarm to the model management layer to initiate a new ML project.

2.3.2 AI/ML Management for 5G Systems

The increased utilization of AI and ML has underscored its significance in the 5G mobile networks sector and future generations. In September 2023, the 3GPP published

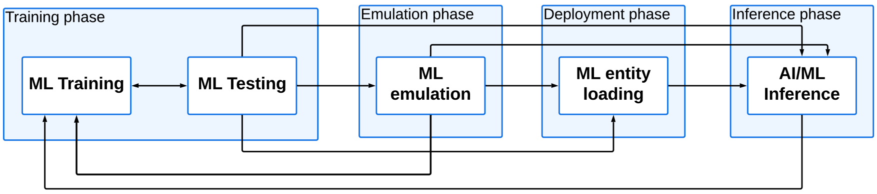


Figure 2.7: Operational workflow of AI/ML [1].

AI/ML management definitions for 5G definitions systems [1].

A 3GPP working group proposed a generic operational workflow for AI/ML for an ML entity in the normative phase, as illustrated in Figure 2.7. Notably, the storage and provisioning of pre-trained ML models for NWDAF are beyond the scope of 3GPP [3].

The workflow involves four main phases: (I) training phase, (ii) emulation, (iii) deployment, and (iv) inference. The training phase comprises training and testing of ML algorithms. The emulation phase executes the created model and evaluates its performance before applying it to the network. It is optional. The deployment phase is when we load the model for inferences. It uses the model in the interface phase.

2.4 Anomaly Detection

Data analytics applies statistical and logical techniques to evaluate data from specific processes. The main objective of the practice is to extract useful information from data [9]. Anomaly detection is a data analytics technique that identifies rare events or observations that may raise suspicion for being statistically different from the rest of the data. Various factors, such as errors, fraud, cyber-attacks, or natural events, can cause anomalies. In this work, we perform anomaly detection using ML techniques to learn the normal pattern of the data and then identify anomalies that deviate from this pattern.

2.5 Chapter Conclusion

This chapter introduced fundamental concepts for understanding the master thesis. We aim to analyze data in the 5G network using AI, which are used in a SBA consisting of various network functions. In 5G SBA networks, the function responsible for analyzing data is the Network Data Analytics Function. We study and present details about NWDAF.

Chapter 3

Literature Review

Data analytics of networks has been a constant activity for proactive network maintenance. The AI resources expect to analyze 5G networks to obtain data and results that promote automatic network improvement actions. In this sense, 3GPP has established standards, and several works offer implementation suggestions for a 5G network data analytics function.

The search string “NWDAF machine learning 3GPP 5G architecture -wireless” entered on the site scholar.google.com.br returned 77 results. Twenty-two papers meet the inclusion criteria in Table 1.1 of the systematic review methodology. We observe aspects regarding architecture to support data analytics algorithms, container technology, software usage in the architecture, ML technology, the possibility of algorithm alteration, evaluation metrics, and execution in a 5GC.

Next, we present twenty-two papers associated with our motivation and objective and prepare the papers in two subjects: (i) NWDAF Frameworks and (ii) 5G Data Analysis with AI. We chose two subjects because some papers focused on applying AI and others on designing a Framework NWDAF. Each subject has a summary Table with four features associated with our master thesis objective: (i) Framework or Platform; (ii) 3GPP; (iii) Possibility to change algorithms; (iv) Demonstration; and (v) with AI. Also, empty cells mean that paper does not have the characteristic that “yes” cells have. We describe each Table column below.

- **Framework or Platform:** works presents an architecture or experimental execution in technology as Kubernetes, Cluster, Docker, or similar;
- **3GPP:** proposed works according 3GPP TS.
- **Possibility to change algorithms:** works presents flexibility and steps to change algorithms whether change the analytics use case. i.e., the proposed solution to analytics can be used for many use cases, including future use cases;
- **Demonstration:** works presents execution analytics or analytics steps or how to use algorithms and the proposal;
- **With AI:** works presents analytics using AI algorithms.

In summary, the Chapter is presented as below: Section 3.1 presents works related to the 5G network data analytics function; Then, 3.2 presents 5G data analytics with ML; Finally, Section 3.3 concludes the chapter.

3.1 NWDAF Frameworks

The authors in [48] present an NWDAF approach for network traffic analytics in the context of 5GC signaling. The work presents the development of an NWDAF prototype and network traffic data analytics to explore the main interactions between NFs using unsupervised learning (K-means) to cluster the primary interactions. The authors implement the 5G using Open5GS[1], Evolved Packet Core (EPC), and UERANSIM[2]. The software used in the NWDAF architecture were Hyper-V[3], Apache Kafka[4], and MongoDB[5]. The mean, maximum, and standard deviation of packet length and total packet count were evaluation metrics of the NWDAF. The approach does not consider changing algorithms for data analytics from different contexts.

The work in [18] develops an NWDAF for network traffic data analytics and discusses its influence on MANO activities. Open5GS and UERANSIM form the analytics environment. The execution of NWDAF involves evaluating its operation through packet counting using protocol and packet size. The work's architecture does not incorporate software, and it does not utilize ML techniques. It does not present known evaluation metrics. Furthermore, the work does not allow changing algorithms for data analytics in other contexts.

The authors in [7] apply data analytics in the lifecycle management of a network slice for load prediction and network resource scaling. The architecture consists of the following software: open-source MANO (OSM)[6], Flexran controller[7], Grafana[8], Prometheus[9], and ElasticMon. The 5G environment used was OpenAir-Interface (OAI)[10]. The authors implemented the Random Forest (RF), Catboost, and XGBoost algorithms for DDoS detection and Long short-term memory (LSTM) for load prediction. The evalu-

[1] https://open5gs.org

[2] https://github.com/aligungr/UERANSIM

[3] https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/hyper-v-technology-overview

[4] https://kafka.apache.org

[5] https://www.mongodb.com/pt-br

[6] https://osm.etsi.org

[7] https://mosaic5g.io/flexran/

[8] https://grafana.com

[9] https://prometheus.io

[10] https://openairinterface.org/oai-5g-core-network-project/

ation metric measured the accuracy of the algorithms. This work presents a single metric for evaluating the algorithms and does not present the possibility of changing algorithms for different analytics use cases.

The authors in [17] implement the NWDAF in AWS services (Lambda and API gateway) and Amazon DynamoDB[11] for network traffic data analytics of a UE in the signaling context and network efficiency. The authors used meta-learning (LSTM, GRU, RNN) to reduce network signaling overhead. In the simulation, the authors prove that it can generalize the proposal to devices with different types of traffic and effectively reduce signaling. However, the work does not present the execution of the algorithms in a 5GC, nor does it present an architecture that allows changing algorithms for data analytics and new analytics use cases.

The authors in [41] present an implementation of NWDAF using Free5GC[12] for the 5GC and UERANSIM for RAN and UE. The result presented the operability verification of NWDAF. Although the code is available on Git, the work does not present an execution in an analytics use case with evaluation metrics and ML techniques.

The authors in [8] present an assisted data analytics mechanism for autonomously managing the end-to-end (E2E) network slice lifecycle. The authors use an Intent-based networking (IBN) platform, NFV orchestrator (NFVO), MANO (OSM), FlexRAN controller (a RAN controller), and a monitoring and data collection mechanism in an OpenAirInterface 5GC. Data analytics results in automatic resource scaling and detection and mitigation of DDoS attacks. The algorithms used for slice quality analytics were GBM, XGBoost, and Catboost. The authors use an attack dataset, the KDD-CUP DDoS [34], and a synthetic attack dataset for training. The trained model detects network anomalies and reports to the Intent-based networking (IBN) platform to execute the mitigation policy and interrupt that flow. To test the stability of the presented system, the authors performed tests on iPerf[13]. It validates the proposed model prediction results on the test dataset through RMSE, MAPE, MAE, MSE, and R2. The presented analytics mechanism does not allow changing data analytics algorithms and new analytics use cases.

The authors in [43] implement an NWDAF with the Analytical Logical Function (AnLF) component and an NWDAF including the Model Training Logical Function (MTLF) component. The implementation used Openapi-generator-cli[14] and Flask[15]. When service consumers request NWDAF containing AnLF with the event ID for the first time, it cannot immediately return analytical results. It is because the AnLF cannot independently create ML models. Initially, there are no ML models to process analytics requests. Still, an experimental study on the analytics and provisioning services of ML

[11] https://aws.amazon.com/pt/dynamodb/

[12] https://free5gc.org

[13] https://iperf.fr

[14] https://github.com/OpenAPITools/openapi-generator

[15] https://flask.palletsprojects.com/en/3.0.x/

models demonstrates the feasibility of the NWDAF test environment. The work does not include the type of analytics data, execution of analytics in a 5GC, or the use of ML techniques.

The work [30] presents two optimization scenarios in the O-RAN use case, namely a predictive model of cell load using LSTM and (ii) an energy-efficient-oriented model using distributed Deep Reinforcement Learning (DRL) models. The data used are available on Github[16]. After quantitative training and validating the AI/ML models, the authors present a general workflow for building, delivering, and evaluating the AI/ML model. The evaluation involved using Mean Squared Error (MSE Loss) graphs per training epoch applied to different learning rate values. The architecture has software Prometheus, Grafana, HTTP API, Airflow[17], and Docker[18]. The architecture allows algorithm changes only for the data analytics use case for RAN efficiency and channel improvement.

The authors in [23] present demonstrations using the architecture of Capgemini Engineering's NWDAF solution for an End-to-End 5G. First, RAN monitoring uses a FlexRIC xApp[19]; then, Capgemini Engineering's Network Data Analytics function collects metrics from the 5GC. Third, Capgemini Engineering's NetAnticipate AI/ML mechanism predicts network function load and the Network Slice instance thought data collected from the leading network for analytics. The company markets the solution to serve some use cases for analytics. The authors do not present ML techniques, implementation in a 5GC, and evaluation.

The work in [75] presents a data analytics architecture for network traffic resource prediction. It uses the software Prometheus, Kubernetes[20], and Kafka in the architecture. It implemented traditional time series forecasting techniques for data analytics, such as autoregressive moving averages (ARMA) and vector autoregression (VAR). Functional evaluation analyzed CPU and RAM percentage, input and output packet count, and input and output byte amount. The OpenAirInterface 5GC performs the architecture analytics. The authors claim that changing algorithms and performing analyses for different use cases is possible. However, the work does not present how to change or add algorithms. Implementation and evaluation of ML techniques were also not presented.

The authors in [57] present an analytics and monitoring architecture designed as a scalable, reliable, low-latency, distributed, reconfigurable, and multi-source data aggregation architecture. The architecture uses Kafka, Elastic (ELK) Stack[21] (Data shipper, Beats, Logstash), and Docker software. The execution 5G environment was 5G EVE[22],

[16] https://github.com/sevgicansalih/nwdaf_data

[17] https://airflow.apache.org

[18] https://www.docker.com

[19] https://gitlab.eurecom.fr/mosaic5g/flexric

[20] https://kubernetes.io/pt-br/

[21] https://www.elastic.co/pt/elastic-stack

[22] https://www.5g-eve.eu

5TONIC[23], and data emulation with Sangrenel[24]. Experiments verificate to evaluate CPU(%), RAM, and data writing time. The architecture does not assess open-source 5GC and ML techniques and their respective evaluations.

The authors [12] present a flexible and scalable analytics architecture based on open-source microservices available for use. By designing the Analytics framework, the authors meet diverse requirements from different solutions and their corresponding use cases. The Analytics architecture enables the application of various algorithms easily through different Analytics services. Usage test in the 5GENESIS[25] environment in an NSA 5GC network. The lifecycle management of the ML model uses tools such as Acumos[26] and Kubeflow[27]. The authors use methods like Z-Score and Median Absolute Deviation (MAD) For anomaly detection. Time series maintain synchronicity with time in advanced analytics. Statistical analyses are performed for the validation process using minimum, maximum, standard deviation, and mean. It performed correlation algorithms like Pearson, Kendall, and Spearman to find linear relationships between variables collected during experiments. For feature selection, it uses algorithms like Backward, Recursive Feature Elimination (RFE), and Least Absolute Shrinkage and Selection Operator (LASSO) to identify the most relevant relation variables to a target variable. The prediction uses linear regression, Random Forest (RF), and Support Vector Machines (SVM) algorithms for prediction. Despite the various techniques applied, the architecture does not allow the alteration or addition of new algorithms.

The work [21] explains the Use Case Development Interface (UDI) of NWDAF, which defines protocols between NFs to interact for data exchange and, subsequently, to track and evaluate actions. The 3GPP has no specified standards for model sharing among various vendor environments, and the proposed NWDAF UDI architecture enables the interoperability of different models and connections for all domains[21]. However, the work does not present evaluation, ML techniques, types of analytics data, and analytics execution in a 5GC.

None of the works use container technology in their architecture using all the selected software components we use in this master thesis. Table 3.1 presents aspects of the research works of the systematic review and the framework proposed in this master thesis.

[23] https://www.5tonic.org
[24] https://github.com/jamiealquiza/sangrenel
[25] https://5genesis.eu
[26] https://www.acumos.org
[27] https://www.kubeflow.org

Table 3.1: Comparative aspects of the literature works of the data analytics frameworks.

Works	Framework or Platform	3GPP	Possibility to Change Algorithms	Demonstration	With AI
[48]	yes	yes		yes	yes
[18]		yes		yes	
[7]	yes			yes	yes
[17]	yes	yes			yes
[41]		yes	yes	yes	
[8]		yes		yes	yes
[43]		yes	yes		
[30]	yes			yes	yes
[23]			yes	yes	
[75]	yes		yes	yes	
[57]	yes		yes	yes	
[12]	yes	yes		yes	yes
[21]		yes	yes		

3.2 5G Data Analytics with ML

The authors in [39] present a distributed data analytics architecture consisting of the root NWDAF and multiple local NWDAF in 5GC networks. The NWDAF structure is distributed and adapted for federated learning (FL) in 5G. Edge NWDAF creates local models, and central NWDAF builds global models by aggregating local models. It enables only limited data from specific regions or NFs to be collected and analyzed to reduce network resource consumption and increase data security significantly. The authors do not present an analysis of the use of software and the data types. Additionally, the architecture is not tested in a 5G environment and does not apply ML techniques and evaluation methods.

The authors [63] explore new features in 5G networks to estimate QoS based on network traffic data. For this purpose, they implement the AF that can be customized by third-party applications, such as YouTube and the NWDAF, to obtain data from the 5GC and third parties. SimuLTE[28] was used to simulate an LTE/5G network. It uses the Least Absolute Shrinkage and Selection Operator (LASSO), Location Registration and Recognition (LRR), Kernel ridge regression (KRR), and Support Vector Regression (SVR) algorithms. Training and cross-validation were applied five times for each algorithm. A model evaluation uses the root mean square error (RMSE) metric. The proposed implementation does not present an architecture with software that enables the execution of other algorithms, being a proposal limited to the data analytics use case in the QoS

[28] https://github.com/inet-framework/simulte

context. Additionally, it does not present execution in an open-source 5G environment.

The authors [14] integrate data-driven intelligence through Graph Neural Networks (GNNs) and deep learning to support security in virtual 5G networks. The authors in [14] model interactions between microservices deployed in the Ku-Kubernetes target application through a network graph (dependency network graph), illustrating cloud interaction almost in real-time. It evaluated the implemented algorithms for their accuracy and recall. The solution does not present software in its architecture or allow algorithm alteration for other analytics use cases. Additionally, it does not mention implementation in a 5GC.

The authors [40] present an NWDAF architecture, where various services can share one or more models for the analytics use case of 4G/5G network efficiency. Any Data Lake data trains each model. The data refers to mobility signaling from a single AMF. Clustering algorithms, Markov prediction, recurrent neural network (RNN), and LSTM model were implemented and evaluated for accuracy and recall when trained with three 4G network datasets. The architecture does not consist of a set of software and does not allow the alteration of algorithms for new analytics use cases.

The authors in [60] describe a project for managing virtualized 5G networks. The analyzed data refer to network traffic, specifically packets related to UPF, SMF, and RAN functions. It uses the software Docker, Open-source MANO (OSM), Open Network Automation Platform (ONAP), Network Configuration Protocol (NETCONF) interface, and Acumos AI in the architecture. AI techniques perform in three analytics use cases: UPF Anomalous Behavior Identification (eXtreme Gradient Boosting (XGBoost)), 5G Cell Traffic Load Prediction (Recurrent Neural Networks (RNN), LSTM, and Linear Regression (LR)), Closed-Loop Automation for SMF Resource Usage (AutoRegressive (AR), Vector AutoRegressive (VAR), and LSTM). The authors execute the project on 5G MaSoN and evaluate the CPU and RAM percentage, F1-score, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Mean Squared Error (MSE) of the models. The system is designed to be flexible and can be extended for other use cases in a 5G network, but the author did not explain if other people could include new algorithms or only the author could consist of it.

The authors in [62] present a demonstration of a Quality of Service (QoS) Monitoring and Prediction Platform (QMP) for proactive detection and mitigation of QoS degradation in slice network operation. The ML algorithms use LSTM, RNN, and Support Vector Regression (SVR). Precision metrics evaluate the models. The software used in the platform architecture were Kubernetes, Kafka, Prometheus, and Grafana. The authors use MLOps paradigms to deploy and maintain ML models reliably and effectively in production. It implements the Content Delivery Network (CDN) platform in a 5G-enabled cloud installation. The architecture did not allow the alteration of algorithms for new analytics use cases and did not integrate with a 5GC.

Based on network latency, throughput, and the number of devices, the authors in [13] present a neural network-based billing solution and recurrent neural networks to predict the billing plan and slice network load in the context of the slice as a service. Model accuracy evaluates the solution. The authors do not present an architecture or a set of software that allows the alteration of algorithms for data analytics in different contexts. Also, the solution does not present execution in a 5G network.

The authors in [35] present a distributed model training architecture to improve NF data localization, enhance security, reduce control overhead during model training, shorten training time, and increase the accuracy of models trained due to local tests in natural environments. Crucial network functions to install the NWDAF, where a centralizer receives data from all local NWDAF. They cited Deep Neural Networks as a suitable AI technology for distributed training. There was no use of software in the architecture and execution of NWDAF in a 5G network. The author does not present the data types for the analysis and evaluation of the proposal. Also, it is impossible to change algorithms for new analytics use cases.

Using a Fuzzy Logic algorithm, the authors in [72] predict channel mobility context and congestion level calculation for better QoE. Video-rate, freezing, SINR, bandwidth availability, 5G cell availability, and handover quantity data are analyzed. The authors assume that NWDAF obtains user equipment location information through location servers. The authors do not present an architecture, a set of software, that allows the alteration of algorithms for data analytics in different contexts. Additionally, they did not use algorithm evaluation metrics, and they did not execute the solution was not executed in a 5G network.

A few works that presented data analytics using container technology in the framework proposed in the thesis are their architecture, ML algorithms, execution of data analytics in a 5GC, and/or support for altering algorithms for different analytics use cases. Some works did not demonstrate the proposed 5G network data analytics and the list of software integrated into the container. Table 3.2 presents aspects of the research works of the systematic review and the framework proposed in this master thesis.

3.3 Chapter Conclusion

This section presented the works related to this dissertation. It presented tables with essential aspects of the works. None of the works proposes a 5G network analytics function that uses container technology, open-source software, and ML techniques used in this master thesis. Additionally, none of the works presented a user-friendly interface that allows the inclusion or alteration of analytics algorithms.

Table 3.2: Comparative aspects of the literature works of the 5G data analytics with AI

Works	Framework or Platform	3GPP	Possibility to Change Algorithms	Demonstration	With AI
[39]					yes
[63]		yes		yes	yes
[14]				yes	yes
[40]		yes		yes	yes
[60]	yes	yes	yes	yes	yes
[62]	yes	yes		yes	yes
[13]		yes			yes
[35]		yes			yes
[72]		yes			yes

Chapter 4

Development

The Chapter outlines the progression of the master thesis.The framework allows ML algorithms to analyze 5G network data and add new algorithms. The development comprises studies of ML algorithms and the conception of the framework. The ML algorithms studies develop some unsupervised and supervised learning algorithms for comparison and developing knowledge in ML. The framework conception presents requirements specifications, architectural design, and framework implementation.

4.1 Machine Learning Algorithms Studies

The section presents studies of some ML algorithms. This studies was crucial part for obtaining knowledge in ML algorithms. The complete studies can be accessed in [69] [53]. In addition, the study can refer to the first phrase (training phrase) in the 3GPP generic operational workflow for AI/ML presented in Section 2.3.2 because we use the ML before applying in a 5GC [1].

4.1.1 Supervised Learning

We present a comparative study of the different supervised ML algorithms using the SCD-2 Brazilian satellite dataset. Satellite space missions generate two types of data: mission data (coming from the payload) and maintenance data (coming from the service module). Mission data is sporadic and sent in large volumes to ground stations. Maintenance data is generated continuously in smaller volumes [36], which concerns mainly the "health" state of the satellite.

The mission of SCD2 is to collect meteorological data from 750 data collection platforms spread across Brazilian territory and relay them to an earth station in the city of Cuiabá, in the state of Mato Grosso. SCD2 is an experimental project developed by the MECB program (Missão Espacial Completa Brasileira). Launched on October 22, 1998, its useful life was estimated at up to two years, which has already been exceeded, and has been in operation for over 23 years [20].

The study methodology consists of the following steps: (i) Data analytics; (ii) Selection of attributes for application in the models; (iii) Application of ML algorithm; (iv) Evaluation of algorithms; and (v) Results and analytics. We present the steps (iv) and (v) in the Evaluation Chapter 5.

The algorithms chosen for performance verification and comparison were: **Decision Tree (DT)** [67]; **Random Forest (RF)** [46]; **Support Vector Machine (SVM)** [33] [50]; **Bagging Regressor (BR)** [15]; **K-Nearest Neighbors (KNN)** [27]; and **Multi-Layer Perceptron (MLP)** [55]. Details about the work can be seen in the paper at [69].

4.1.2 Unsupervised Learning

A comparative analytics between unsupervised learning algorithms for anomaly detection used NSL-KDD dataset [68]. The dataset is highly unbalanced and used as a discriminative tool in network-based anomaly detection. Due to the lack of public datasets for network-based intrusion detection systems (IDS), we chose the NSL-KDD, a practical reference dataset to help researchers compare different intrusion detection methods. Furthermore, the dataset is rich in detail, highly unbalanced, and easy to download.

The study methodology consists of the following steps: (1) viability study of feature selection and dimension reduction, (2) hyperparameter selection, (3) algorithm selection, and (4) algorithm evaluation. We present the step (4) in the Evaluation Chapter 5. In the viability study of feature selection and dimension reduction, we executed four experiments to check the efficacy or viability of feature selection (using Pearson's correlation) and dimension reduction (using PCA) working together or isolated. More details about the four experiments are below.

- **E1:** The first experiment applies feature selection and reduction. Output occurs after data reduction.

- **E2:** The second experiment does not apply feature selection and reduction techniques. As a result, data output occurs immediately after data preprocessing.

- **E3:** The third experiment applies feature selection techniques only. Output occurs after feature selection.

- **E4:** The fourth experiment applies only data reduction techniques. As a result, the data output occurs immediately after the reduction techniques.

The unsupervised algorithms were **Isolation Forest**, **Local Outlier Factor** (LOF) [16], **Elliptic Envelope** (EE) [61], **Stochastic Gradient Descent One-class Support Vector Machines** (SGD or SGD One Class SVM)[37], **Isolation Forest** (IF) [44]. Details about the work can be seen in the paper at [53].

4.2 The Framework Conception

The master's thesis conceives a computational solution to data analytics on 5G designed as a framework. A framework is a set of cooperating classes that make up a reusable software project [28]. Our flexible framework makes it possible to change analytics use case algorithms. It provides a user-friendly environment, enabling users to perform data analytics on the 5G network without the need to delve into the TS of NWDAF or understand the full set of requests and APIs involved.

A functional framework capable of including algorithms to analyze 5G network data and present analytics information on a real-time monitoring dashboard. In addition, the framework collects and storage data, algorithms, and ML models. Its architecture uses open-source software, and it is easy to implement. We conceive the proposed 5G network data analytics framework from requirements specifications, architectural design, and implementation. The conceive detail is below.

4.2.1 The Requirement Specification of the Framework

The requirement aims for a functional framework capable of including algorithms to analyze 5G network data and present analytics information on a real-time monitoring dashboard in acordding 3GPP NWDAF. Each software must be chosen according to the 3GPP NWDAF components so that the framework has all the functionalities expected by standardization.

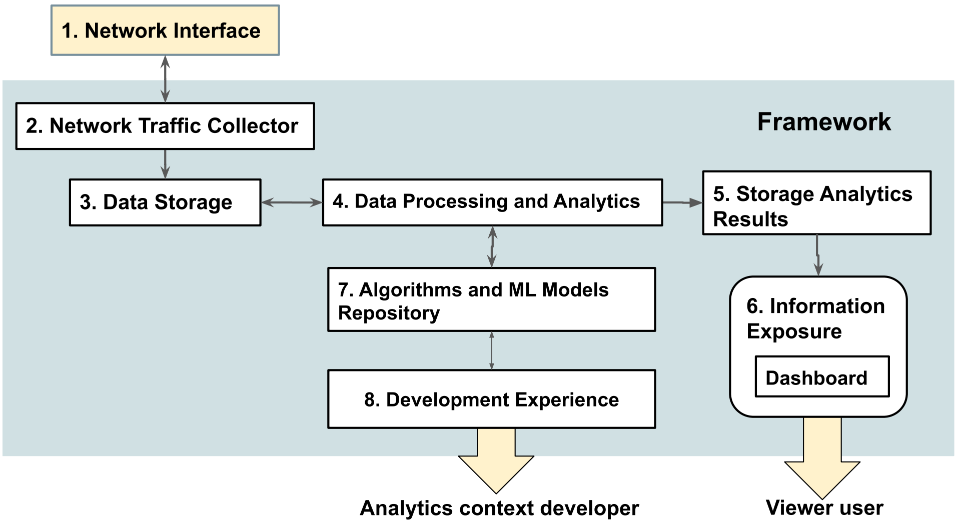

1. Network Interface
Framework
2. Network Traffic Collector
3. Data Storage
4. Data Processing and Analytics
5. Storage Analytics Results
6. Information Exposure
Dashboard
7. Algorithms and ML Models Repository
8. Development Experience
Analytics context developer
Viewer user

Figure 4.1: Framework network data analytics architecture.

The framework collected data in pcap format (standard format for storing captured network packets). We chose this data format because it is widely used in the literature for network data analytics. A network traffic analyzer software collects the pcap data format. The storage and analytics components must have volume, variety, velocity, complexity, and interoperability as characteristics because the framework must be prepared to operate in 5G and future networks with different types of data that may be considered in the future. These characteristics align with Big Data because our framework must function as a data processing environment similar to Big Data systems. In addition, an interface for editing algorithms must be included to facilitate the analytics algorithms develop.

The complexity of the communication and data collection of other 5G network functions can be as intricate as the network size and the number of network functions in the 5GC. The framework performs analyses for one analytics use case at a time. If changing the analytics use case is necessary, we develop new algorithms and restart the framework.

4.2.2 The Framework Architecture Design

Given the requirements specification, we design the framework architecture for analyzing 5G network data. Figure 4.1 presents the framework architecture, and below are the functions of the architecture components.

1. **Network Interface:** is the interface that receives or sends network traffic packets

from the computer to the internet. The architecture connects to this interface belonging to the 5G network core;

2. **Network Traffic Collector:** responsible for collecting network traffic data. Data is collected at the Network Interface and stored in the Data Storage. The format of the collected data is .pcap;

3. **Data Storage:** repository of data collected by the Network traffic collector. It can store and provide data at any time and in any quantity. The stored data include (i) data for ML model training, (ii) collected data to be processed, and (iii) processed data to be analyzed;

4. **Data Analytics:** responsible for analyzing network data using data stored in the Data storage. It executes algorithms and stores trained ML models in the Algorithms and ML models repository.

5. **Storage Analytics Results:** stores the results of data analyses performed in the Data analytics.

6. **Information Exposure:** presents information from the Storage analytics results through a dashboard for user visualization.

7. **Algorithms and ML Models Repository:** storage of algorithms and analytics models ready to execute. The algorithm is in .json format in the framework installation folder.

8. **Developer Experience:** interface for including and editing algorithms in the framework by the analytics use case developer.

Figure 4.2 presents framework architecture with 3GPP NWDAF logical functions. The Network interface is a **Data souce** (NF, AF, UDM, and OAM). The Network Traffic Collector and Data storage is a **Data Collection Coordination (and Delivery) Function** (DCCF). The data processing and analytics is an **Analytics logical function** (AnLF). The storage analytics results are an **Analytical Data Repository Function** (ADRF). The information exposure is a **Data analytics output**. The Algorithms and ML model repository is a **Model training logical function** (MTLF). Development experience and Dashboard are not part of the 3GPP specification.

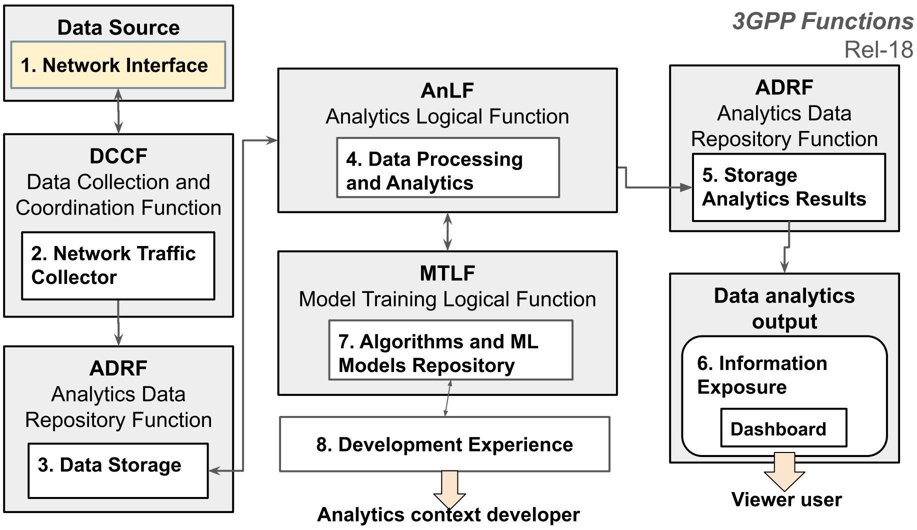

3GPP Functions
Rel-18
Data Source
1. Network Interface
DCCF
Data Collection and Coordination Function
2. Network Traffic Collector
ADRF
Analytics Data Repository Function
3. Data Storage
AnLF
Analytics Logical Function
4. Data Processing and Analytics
MTLF
Model Training Logical Function
7. Algorithms and ML Models Repository
8. Development Experience
Analytics context developer
ADRF
Analytics Data Repository Function
5. Storage Analytics Results
Data analytics output
6. Information Exposure
Dashboard
Viewer user

Figure 4.2: 5G network data analytics system architecture related to NWDAF components presented by 3GPP.

4.2.3 The Framework Implementation

The framework implementation presents the chosen software and its integration by the framework architecture presented previously. In addition, we detail the communication between the 5G functions and the framework. The implementation uses container technology and The software modules of the framework used in the container implementation are called components. We base the choice of components on the requirements and objectives defined in the requirements specification. So, the proposed framework is in the form of a container of integrated components.

First, Docker, a container technology, is chosen to implement the data analytics framework because it guarantees characteristics such as portability, resource efficiency, scalability, and swift development. Next, we choose TCPdump, Hadoop Distributed File System (HDFS), Apache Zeppelin, Apache Flink, PostgreSQL, and Grafana components. Figure 4.3 presents the framework's architecture with the integrated software components. These software were chosen because they correspond to the framework requirements presented and the TS of the 3GPP NWDAF components.

Second, we implement the Network Traffic Collector using **tcpdump**[1]. **tcpdump** is a widely used network packet capture tool commonly used for network monitoring and analytics (network sniffer). It provides a packet-filtering language for capturing traffic of interest.

Third, we implement the Data Storage with **Hadoop Distributed File System (HDFS)**, which is a distributed file system that handles large datasets running on

[1] https://www.tcpdump.org

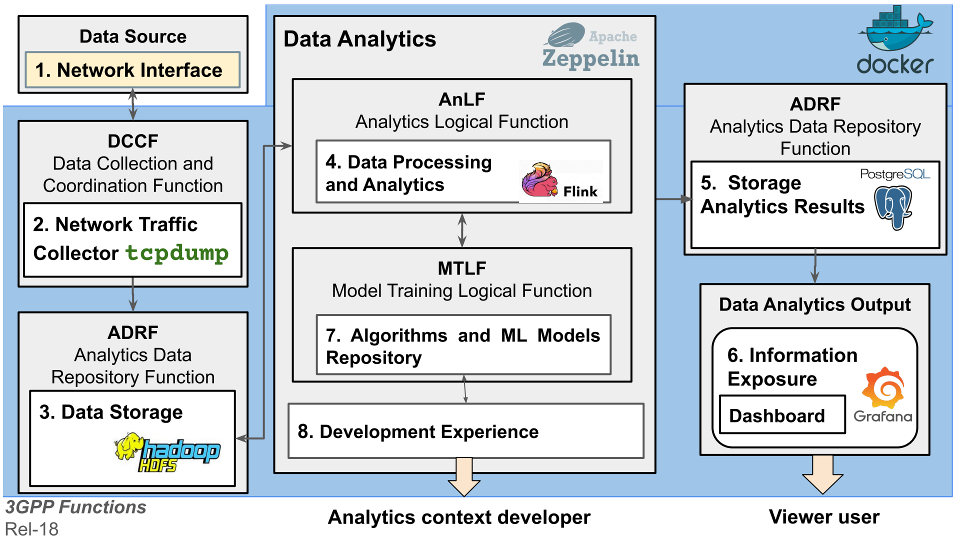

Data Source
1. Network Interface
DCCF
Data Collection and Coordination Function
2. Network Traffic Collector tcpdump
ADRF
Analytics Data Repository Function
3. Data Storage
hadoop HDFS
Data Analytics
Apache Zeppelin
AnLF
Analytics Logical Function
4. Data Processing and Analytics
Flink
MTLF
Model Training Logical Function
7. Algorithms and ML Models Repository
8. Development Experience
docker
ADRF
Analytics Data Repository Function
5. Storage Analytics Results
PostgreSQL
Data Analytics Output
6. Information Exposure
Dashboard
Grafana
3GPP Functions
Rel-18
Analytics context developer
Viewer user

Figure 4.3: Architecture and components of the 5G network data analytics framework.

commodity hardware [32]. Hadoop is capable of processing high-density data [45].

Fourth, the Data Analytics module contains two components, **Apache Zeppelin** and **Apache Flink**. **Apache Zeppelin** is a web-based notebook for interactive data analytics and collaborative data-driven documents with SQL, Scala, Python, R, and more [76]. It supports multiple users simultaneously and has a user-friendly interface. **Apache Flink** is an open-source real-time and batch processing framework. It can run on all cluster environments to perform computations at memory speed and any scale [24]. Flink 1.10 or higher is compatible with Apache Zeppelin. **PyFlink** was used as the Python entry point for Flink in Apache Zeppelin.

Next, the relational database management system **PostgreSQL** was chosen for the **Storage analytics Results**. **PostgreSQL** is a free and open-source database emphasizing extensibility and SQL compliance [59].

Finally, the Dashboard presents the Data Exposure for user visualization. The chosen software is Grafana. **Grafana** is an open-source, multi-platform web analytics and interactive web visualization application. It provides tables, graphs, and alerts for the web when connected to supported data sources. It is expandable through a plugin system [42].

About the analytics and exposure functionality, Apache Zeppelin sends the data analytics results to PostgreSQL. PostgreSQL stores the results, i.e., the obtained information, for Grafana to capture and present on its Dashboard. As a framework developed using container technology, it can be easily explored for scalability. As a framework that can be integrated into the 5GC, its security is expected to be the same as that applied to the 5GC.

An important point to be clarified below is the communication of the framework with the 5GC functions. According to 3GPP TS 23.501, the functions of the 5GC network communicate through 3GPP standardized APIs, such as the REST API. These APIs communicate with each other by sending packets over the network, which are transported via network interfaces. The 5GC utilizes the computer network interface on which it is installed as a communication bus between functions to transmit network packets. During the core installation, each function of the 5GC receives an IP address as identification within the core's internal network. Thus, the functions communicate by transmitting packets via the network interface. It is important to note that the transmitted packets pertain to the Control Plane (CP) and the network User Plane (UP). The framework collects the packets that transit on the network interface, and therefore, it is relevant to capture the network packets for analytics by the proposed framework. This way, the framework obtains all network packets without API integration.

4.3 Chapter Conclusion

The chapter described the development of the 5G network data analytics framework. First, we studied some ML algorithms, specifically supervised and unsupervised learning. Next is the framework conception. Then, we specified requirements, selected components, designed architecture, and implemented components. With that, we concluded the chapter.

Chapter 5

Evaluation

The Chapter critically assesses the progress made in developing the master's thesis, including the ML algorithms and framework conception. The evaluation chapter has three parts: First, we present supervised and unsupervised learning evaluations provided by ML algorithm studies (Section 5.1). We use the best ML algorithms evaluated in an analytics use case. The analytics use case is part of the framework evaluation. Second, we present a framework evaluation that includes an analytics use case, experiments detail with evaluation steps, metrics, and results (Section 5.2). Third, we present the framework and Free5GC integration (Section 5.3). Finally, we conclude the Chapter, Section 5.4.

5.1 ML Algorithms Studies Results

Comparative ML studies are essential before determining the ideal algorithms for analytics. In this section, we present the main results of the evaluation of supervised and unsupervised algorithms. Supervised learning uses mean absolute error (MAE), and root mean squared error (RMSE) evaluation metrics. Unsupervised learning uses F1-score and AUC evaluation metrics. The ML algorithms studies were conducted in an exploratory aimed at advancing knowledge in the field of AI. The ML algorithms with excellent performance were used in the framework evaluation. Each evaluation results are presented below.

5.1.1 Supervised Learning

The amount of errors evaluates the performance of ML algorithms for regression models. We use two evaluation metrics: the **mean absolute error (MAE)**, which mea-

Table 5.1: Average Result of the Evaluation Measurements of the Experiments and Average Processing Time in Seconds for Each of the Algorithms on the Test Data.

Algorithm	MAE	RMSE	R2	Time (s)
DT	0.1581	0.2424	0.9751	**0.13**
RF	0.1490	0.2307	0.9756	2.46
KNN	0.4235	0.5895	0.8358	0.32
SVM	0.1435	**0.1968**	0.9841	0.85
BR	**0.1420**	0.1974	0.9842	0.58
MLP	0.1825	0.2946	**0.9877**	6.81

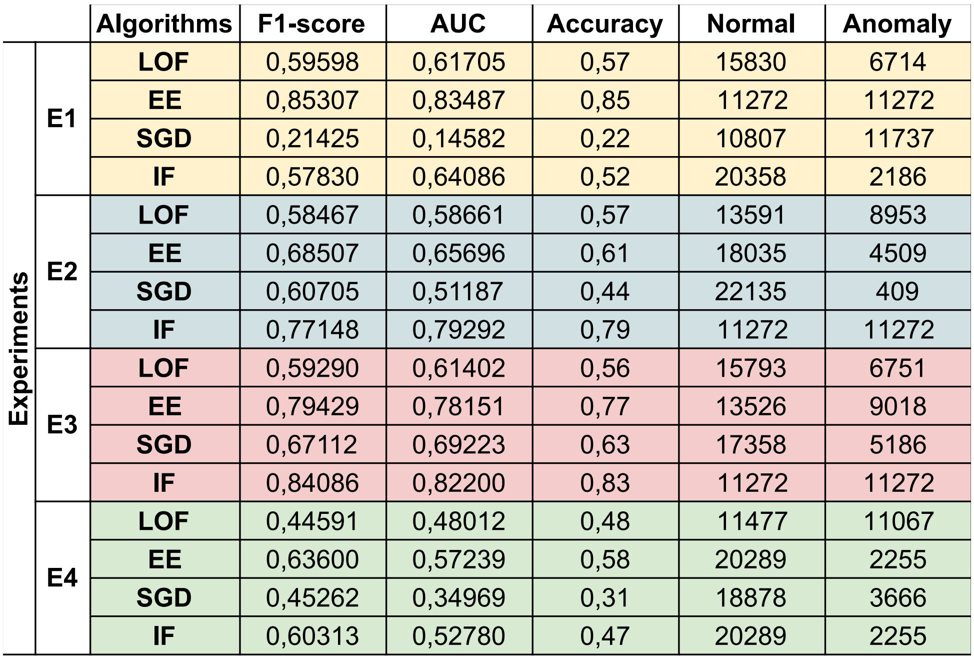
sures how close the predictions are to the actual results, so the lower the value, the better the result; the **root mean squared error (RMSE)**, which calculates the root mean squared error between real values and predictions. This metric disregards the difference between over-predicting and under-predicting when it squares the difference. We compare the outputs of the algorithms and the results of the evaluations. The evaluation methods demonstrate the accuracy of the algorithm when performing the prediction. Table 5.1 presents the average of the algorithm evaluation results and the average processing time concerning the test data (two thousand samples). The best results are bold-highlighted.

The algorithms, on average, obtained very close results, with only the KNN showing a more significant deviation due to its mathematical simplicity. The MLP algorithm required more processing and, therefore, more execution time. The RF is an extension of the DT algorithm, being more complex due to the set of trees it presents. It is more reliable concerning the DT because the decision tree suffers from the overfitting problem. As a disadvantage, RF has a much more expressive processing time than DT. The SVM and BR algorithms presented similar results since the BR algorithm implementation uses SVM as a basis in search of better results. These results were possible because the temperature of the main battery correlates with other attributes available in the dataset and the large amount of data available in short periods.

5.1.2 Unsupervised Learning

We analyze the ML algorithms results by evaluating, done according to the chosen ML technique [58]. A complex task is determining evaluation metrics for the results of unsupervised algorithms; in this work, we use dataset labels in evaluation. The algorithm training does not use labels. The metrics used for analytics are F1-score and AUC, both related to confusion matrix, precision, and recall.

Table 5.2: F1-score, AUC, Accuracy, and amount of normal/anomaly data per algorithm in the experiments.



		Algorithms	F1-score	AUC	Accuracy	Normal	Anomaly
Experiments	E1	LOF	0,59598	0,61705	0,57	15830	6714
		EE	0,85307	0,83487	0,85	11272	11272
		SGD	0,21425	0,14582	0,22	10807	11737
		IF	0,57830	0,64086	0,52	20358	2186
	E2	LOF	0,58467	0,58661	0,57	13591	8953
		EE	0,68507	0,65696	0,61	18035	4509
		SGD	0,60705	0,51187	0,44	22135	409
		IF	0,77148	0,79292	0,79	11272	11272
	E3	LOF	0,59290	0,61402	0,56	15793	6751
		EE	0,79429	0,78151	0,77	13526	9018
		SGD	0,67112	0,69223	0,63	17358	5186
		IF	0,84086	0,82200	0,83	11272	11272
	E4	LOF	0,44591	0,48012	0,48	11477	11067
		EE	0,63600	0,57239	0,58	20289	2255
		SGD	0,45262	0,34969	0,31	18878	3666
		IF	0,60313	0,52780	0,47	20289	2255

In this phase, the algorithm's outputs are presented and compared with the labels in the dataset. The evaluation methods demonstrate the accuracy of the algorithm when performing the prediction. The algorithm's results indicate the existence of network anomalies. In this case, the best algorithm is the one that performs the correct behavior prediction compared to the labeled dataset and has the best evaluation in the applied methods. In addition, we calculated the mean and standard deviation of each algorithm for the four experiments performed. In this way, it is possible to visualize the impact of the techniques on each algorithm.

Table 5.2 presents each experiment evaluation results, F1-score and AUC. The results of experiment E1 revealed that the Elliptic Envelope obtained the highest value concerning F1-Score and AUC. However, there was high variability in the results among the different algorithms. Specifically, it found that the feature selection and dimensionality reduction techniques were ineffective with SGD OneClassSVM and Isolation Forest, resulting in lower performance for them. The results were similar in the experiments E3 and E4, which used one of the two techniques. The experiment with only the feature selection technique performed slightly better than those with only the data dimensionality reduction. Notably, the Isolation Forest achieved a value higher than 80% in the evaluation (E3), standing out as the most efficient. Experiment E2, which did not use either of the two techniques, had poorer results than E3 and E4. We suggest applying at least one of the techniques can benefit the algorithm‘s overall performance.

Figure 5.1 presents the result of the mean (AVG) and standard deviation (SD)

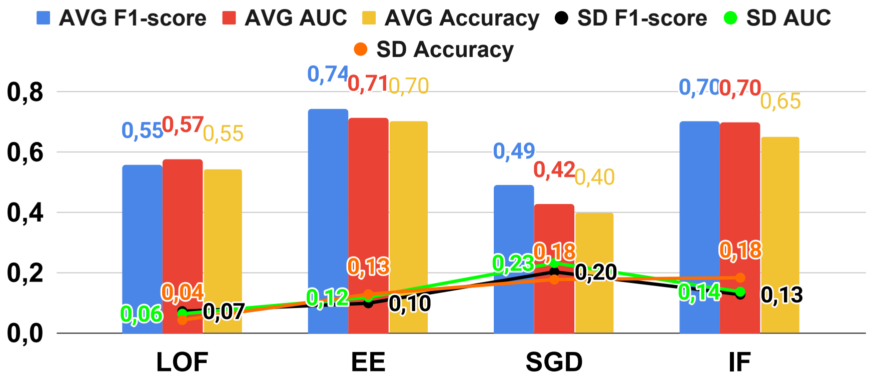


Figure 5.1: Mean (AVG) and standard deviation (SD) of the F1-score, AUC, and accuracy of each algorithm.

of each algorithm. When comparing the means of the results and standard deviation, we observed that the algorithm LOF has the smallest standard deviation. However, the average value of its results was approximately 0.55. Among the four analyzed algorithms, the algorithm EE showed the best evaluation average, while the algorithm SGD recorded the worst. In the experiments, the Elliptic Envelope and Isolation Forest algorithms had exceptional performance, reaching 85% and 83% of accuracy, respectively, while the SGD algorithm obtained the worst result, registering only 22% and 44%.

The LOF exhibited the worst performance when we applied the dimensionality reduction technique (E4) exclusively. The explication is because LOF is on the local density concept, where the k-nearest neighbors define proximity, and their distances can estimate density. When employing the dimensionality reduction technique, we cannot guarantee the preservation of original data in the local density of neighbors. Conversely, the feature selection technique (E3) contributed to faster processing and positively affected the evaluation of this algorithm. EE achieved more favorable evaluation results when we applied dimensionality reduction and feature selection techniques (E1). However, EE displayed the worst performance when we used only the dimensionality reduction technique (E4). SGD achieved better evaluation results when we applied only the feature selection technique (E3). Nevertheless, the worst evaluation occurred when it employed feature selection and dimensionality reduction (E1) techniques. IF obtained the best evaluation results when we applied only feature selection (E3). This algorithm outperformed all the others in all experiments. In this scenario, the algorithm classified 50% of the data as normal and 50% as anomalies. However, IF also recorded the worst F1-score in the experiment 1 (0.5783) and the worst AUC in the experiment 4 (0.5278).

All results emphasize the importance of carefully considering data preprocessing techniques when employing anomaly detection algorithms. Additionally, they highlight the need to investigate further why SGD OneClassSVM and Isolation Forest did not benefit from feature selection and dimensionality reduction to improve their applicability in similar scenarios.

5.2 Framework Evaluation

This evaluation intends to check a framework functionality, not its installation in the 5GC yet. First, we determine an analytics use case to give direction about choosing a dataset for evaluation. Then, we applied the steps to train ML model. Next, we provide details about the experiment. In the final, we presented the framework evaluation results.

The choice of the analytics use case is the step preceding the algorithm's input in the proposed framework. We define anomaly detection in 5G cybersecurity as an analytics use case due to the increase in cyber-attacks and the importance of protecting the 5GC network. Then, we use the best evaluation algorithms in the ML studies as a way to take advantage of the knowledge gained.

Furthermore, we performed experiments to assess our framework functionality, identify operational flaws, store performance metrics, and framework flexibility with the newly added anomaly detection algorithms. The anomaly detection model is assumed to be adequately trained and capable of detecting anomalies. The section presents an analytics use case, details about the evaluation experiments, and experiment results.

5.2.1 An Analytics Use Case: Network Anomaly Detection

The analytics use case development on the framework depends on the ML algorithms. We choose the ML algorithms with the best evaluation in the ML algorithms studies to be used in the anomaly detection use case in the evaluation framework. We choose to use the algorithms of the ML algorithms studies because it is knowledge already obtained and sufficient to correspond to the criteria of the framework evaluation.

The algorithms studied and used in evaluating the framework may or may not be suitable for anomaly detection; however, for this master's thesis, the algorithms with the best F1-score and MAE evaluations in the studies were chosen to leverage the knowledge acquired during the ML studies to test the framework's functionality in an anomaly detection use case. Eventually, the framework can receive other analytics algorithms according to the analytics data.

Furthermore, we consider the best-supervised learning evaluation the Mean Absolute Error (MAE), the smaller the MAE the better the model's performance. On the other side, in unsupervised learning, we consider the best algorithm with the best F1-score; the bigger the F1-score, the better the model's performance. The algorithms **Bagging Regressor (BR)**, **Elliptic Envelope (EE)**, and **Principal Component analytics (PCA)** had the best performances. Details of the analytics use case algorithm development are below.

5.2.1.1 Data Selection

We chose dataset 5GAD-2022 [11] [19] to train the anomaly detection model. The dataset is available at GitHub[1]. The dataset consists of intercepted 5G network data, including normal and malicious traffic data captured by Wireshark in pcap format and made available on GitHub. We chose this data set because its documentation facilitates understanding the data for applying ML techniques. Additionally, we decided on anomaly detection as the analytics use case, and the data refers to cyber attacks on the 5GC.

The 5GAD data [11] were generated in a simulated environment and collected through a Linux machine connected to the internet running a 5GC network implemented with the open-source software Free5GC [25]. The 5GC was connected via Ethernet to another device simulating the User Equipment (UE) and the radio access network using another open-source software called UERANSIM [31].

The structure of the 5GC of the 5GAD-2022 data collection environment is shown in Figure 5.2. The figure presents the functions with the assigned IPs and four network interfaces named as UPFGTPo (green), EN01 (red), LO (blue), and ENP5S0 (orange).

The collected data refers to traffic data in the 5G network and between 5G network functions. There are two types of data: normal and malicious. Normal data are separated by the user's devices, and each attack is separated by malicious data.

The "normal" data falls into two categories: (i) data collected during the simulation execution of a user equipment and (ii) data collected during the simulation execution of two user equipment. In both scenarios, it created a typical user traffic generator to stream YouTube videos, make HTTP/HTTPS requests to commonly visited websites, join video conference meetings with chats, make FTP requests and downloads, and access Samba servers for image, video, and document uploads and downloads.

The "malicious" data relates to ten attacks that fall into three main categories: reconnaissance, denial of service (DOS), and network reconfiguration. These attacks

[1] https://github.com/IdahoLabResearch/5GAD

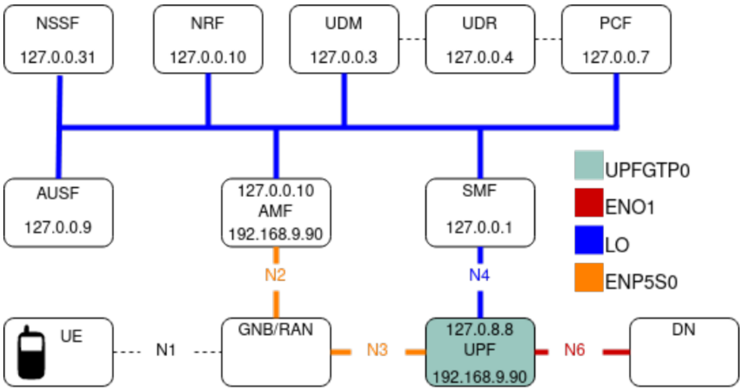


Figure 5.2: Data collection in the 5G environment (5GAD-2022) [19].

assume an intruder has illegitimate communication access to specific 5GC components due to inadequate security configurations. The high-level strategies of these attacks are vulnerabilities documented by Positive Technologies [70], [71], and based on flaws found in Free5GC (each attack used a strategy). The difference between the attacks is irrelevant to this master's thesis, as any attack is an anomaly. Therefore, we refer to the attacks as network anomalies in this work.

5.2.1.2 Data Preprocessing

The 5GAD data are pcap format files. In the preprocessing, to ensure that the information does not influence the training of ML algorithms, such as the source and destination IP addresses, reduce all packets until only the application layer remains. We can do this without fearing the loss of important distinctions between normal and attack packets because the application layer contains all attacks. We discard packages from the training set that do not belong to the application layer. Then, each packet was truncated to 1024 bytes or padded with zeros until it reached a length of 1024 bytes. Truncated packets increased training speed while still including most packet payloads. Most attack packets were less than 1024 bytes. We generated several CSV files and split them into attack (30%) and normal (70%) data files, each separated by user devices.

5.2.1.3 Data Processing

Before using a ML algorithm, we prepare and process a dataset to increase its effectiveness. In this work, we use PCA with three components, meaning three new variables constructed as linear combinations (or mixtures) of the initial variables. Since we did not use feature selection methods, PCA transforms all dataset features into components, consequently reducing the training time of ML algorithms.

5.2.1.4 Machine Learning Algorithms Selection

The algorithms BR (supervised learning) and EE (unsupervised learning) had the best performances and were selected. We use supervised and unsupervised ML at different stages. The unsupervised learning algorithm assigns labels to a dataset, while the supervised ML algorithm generates final models suitable for analytics.

In this stage, we apply processed data in the ML algorithms; `RandomizedSearch` optimizes ML algorithm parameters and trains with a selected dataset. We use unsupervised ML algorithms because they are more suitable for "unlabeled" data than real 5G network data. After labeling and defining patterns in the dataset, supervised algorithms can create models for a specific purpose.

We train the model with supervised ML algorithms and a labeled dataset. The result of the supervised algorithm evaluation is stored, and from this, the framework is ready to use an ideal model for network anomaly detection. The model will receive other network data sets and will be able to verify whether the dataset has anomalies or not.

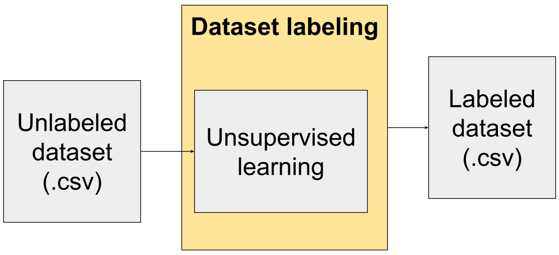


Figure 5.3: Data labeling steps.

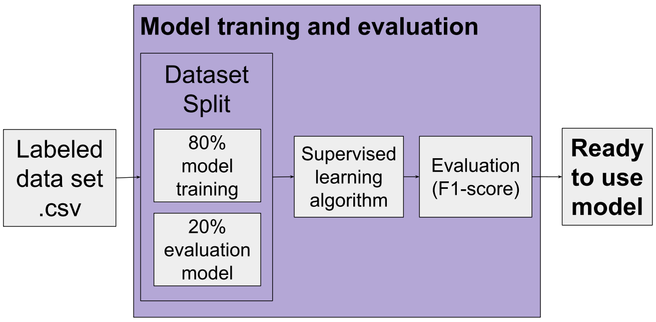


Figure 5.4: Model training steps.

5.2.1.5 Model Training and Evaluation

The training model evaluation verifies whether it can accurately predict or classify new datasets. An evaluation above the expected value signifies the correct training model and its suitability for the framework. We consider the F1-score and AUC greater than 90%. We present the results of the analytics model evaluation in Section 5.2.3.

5.2.2 Experiments

The Framework allows the addition of new analytics algorithms while in operation. With this in mind, we perform experiments in network anomaly detection use cases, specifically in cybersecurity, wherein algorithms have been integrated into the Framework to facilitate its evaluation. The experiments aim to assess the functionality of the newly added anomaly detection framework algorithms, identify operational flaws, and store performance metrics. Through this evaluation, we expect significant insights to enhance the framework and ensure its effectiveness in detecting network anomalies and in future usage contexts.

This evaluation aims to check the functioning of the Framework. First, we observe the correct installation of container components, data collection, and saving of collected data. Next, we collect metrics that infer performance and improve possibilities. The metrics are given below: it obtains numbers of “normal”, “anomalous”, and “total” packets analyzed by the Framework; it obtains the Mean and Standard Deviation (SD) in each experiment; it obtains CPU percentage and RAM (GB) during analytics; it obtains Precision, Recall, Accuracy, F1-score, and AUC in each experiment.

Before the experiment, an injection attack tool is determined. Next, a dataset for injection attacks is defined. This dataset mainly consists of attack traffic packets and regular traffic packets. Finally, a virtual interface network receives only injected traffic packets. Attack details are presented in Section 5.2.2.2.

At the beginning of the experiment, we assume that the anomaly detection model is adequately trained and detecting anomalies, details in Section 5.2.2.1. Initialization commands start the framework and a "stop" command ends it in Docker. The experiment steps are detailed below.

1. The analytics framework is installed five times. Each time you install the Framework, you train the model again. Each installation is an experiment. Each experiment is named EX, where "X" is an experiment number;

2. Each experiment receives five attack injections;

3. At the end of each attack injection, the numbers of "normal", "anomalous", and "total" packets analyzed by Framework are noted;

4. The mean and standard deviation (SD) of the "normal" and "anomalous" packets are calculated in the final of each experiment.

In addition, the experiments were performed in the computer with settings 32 GB of RAM, 480 GB of SSD, Intel® Core™ i7-8550U CPU @ 1.80GHz × 8 processor, and an Ubuntu 22.04.3 LTS operational system.

5.2.2.1 Anomaly Detection in the Framework

We now consider integrated components and network data analytics an adequately trained model. The operation of the Framework's data analytics depends on its integrated components. The workflow of the Framework designed for anomaly detection, considering its model suitable for use, is presented below and in Figure 5.5.

The collection component, TCPdump, is programmed to capture packets at a specific time interval automatically. Upon completing a capture, TCPdump starts another capture in the next instant. Each capture creates a dataset in pcap format in the corresponding folder. The dataset in pcap format storage in HDFS assigns a Universally Unique Identifier (UUID) to the dataset, making it available for analytics algorithms to read. A "verification routine" accesses HDFS to find the unverified dataset. An unverified dataset is one that the Framework's algorithms have not analyzed. When the

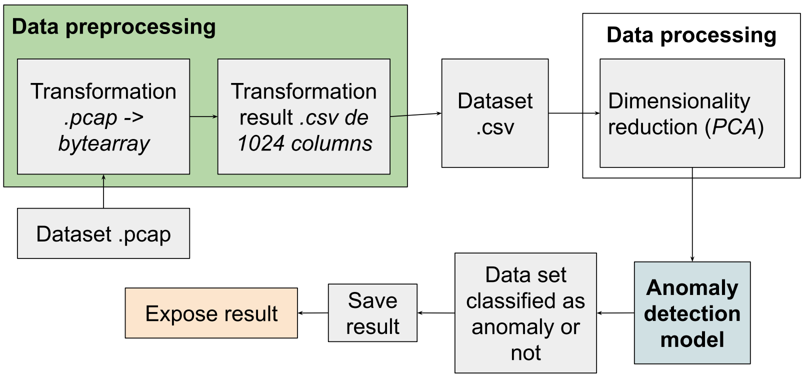

Data preprocessing
Transformation .pcap -> bytearray
Transformation result .csv de 1024 columns
Dataset .csv
Data processing
Dimensionality reduction (PCA)
Dataset .pcap
Anomaly detection model
Data set classified as anomaly or not
Save result
Expose result

Figure 5.5: Workflow designed for anomaly detection.

"verification routine" finds a new "unverified" dataset, the data preprocessing step with the "unverified" dataset begins, and the dataset in pcap format converts into CSV format.

When the dataset becomes a CSV, the anomaly detection model detects the new dataset in CSV format from the "verification routine", initiating the data analytics. The created model receives a CSV dataset and classifies each data row as "anomaly" or "non-anomaly". When the model finishes the analytics, it sends the result to the PostgreSQL database, which stores and provides the obtained information to Grafana. Grafana updates the new information on its dashboard for a specific period and then presents the information obtained in the last check. The dashboard consists of the number of analyses performed, the number of normal packets, and the number of packets with anomalies.

5.2.2.2 The Internal Attacks

The internal attacks are data injected in the network interface created to simulate attacks in our evaluation experiments. We executed malicious attacks on the 5G network using an injection network traffic packet tool and a predominantly malicious network traffic dataset. We create a virtual network interface and use a dataset for the experiments.

After starting the Framework, the .pcap files are injected using Tcpreplay[2].The TCPreplay tool injects network traffic packets referring to malicious attacks, as provided by Coldwell et al. In [19]. Attack .pcap files may contain a few network packets considered "normal". In addition to malicious traffic, the files can include legitimate traffic, which makes the scenario similar to the real attack scenario.

The TCPreplay tool injects traffic packets in the virtual network interface, vr-

[2] https://tcpreplay.appneta.com

br, for evaluation. The vr-br interface only receives network packets injected by the TCPreplay tool, an isolated network environment, without interference from new packets in the assessment. Figure 5.6 presents the experiment environment.

During attack injection, errors may occur when sending some packets or even when sending them randomly. Although this may look like a problem, such a characteristic makes the simulation more realistic, reflecting the imperfections and oscillations that can occur in the real world.

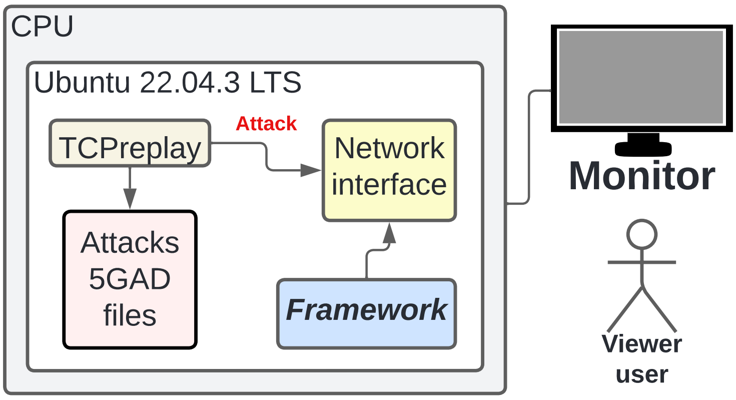
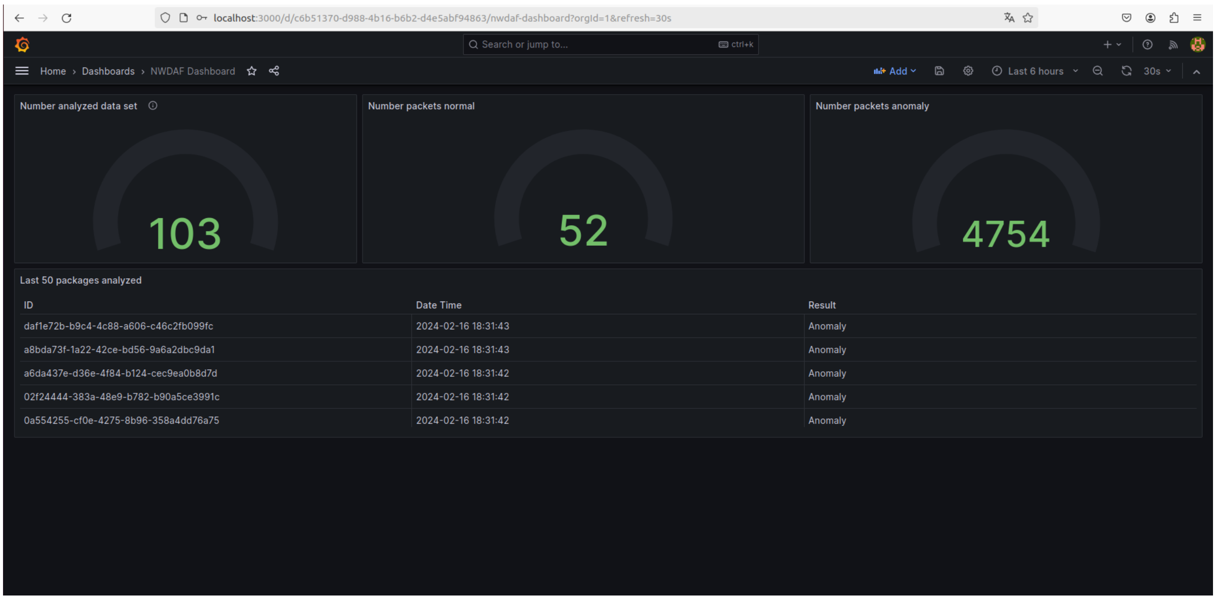


Figure 5.6: Framework evaluation environment.

5.2.3 Results

The framework did not present correct results at the beginning of the experiment. Unsupervised learning labeled part of the data wrong, implying that a model training with the wrong label dataset results in the dashboard being incorrect. Then, the unsupervised learning presented in Figure 5.3 was not considered in the experiment. We used the original dataset label to train the analytics model with supervised learning (Figure 5.4. The analytics model was trained correctly, and a good evaluation was obtained throughout the section.

From the first command, the framework runs for an hour after setting to start collecting packets. This time is required to download the components, install the dependencies, train the ML algorithm, and configure Apache Zeppelin, HDFS, PostgreSQL, and Grafana. The ML algorithm ran, evaluated, and saved correctly in the database. Having the algorithms ready, we trained the anomaly detection model with the prepared dataset. After training, the dashboard opens, and packets and their analytics are captured. Figure 5.7 shows the dashboard during one of the experiments. The dashboard was configured


localhost:3000/d/c6b51370-d988-4b16-b6b2-d4e5abf94863/nwdaf-dashboard?orgId=1&refresh=30s
Search or jump to...
ctrl+k
Home › Dashboards › NWDAF Dashboard
Add
Last 6 hours
30s

Number analyzed data set: 103

Number packets normal: 52

Number packets anomaly: 4754

Last 50 packages analyzed

ID	Date Time	Result
daf1e72b-b9c4-4c88-a606-c46c2fb099fc	2024-02-16 18:31:43	Anomaly
a8bda73f-1a22-42ce-bd56-9a6a2dbc9da1	2024-02-16 18:31:43	Anomaly
a6da437e-d36e-4f84-b124-cec9ea0b8d7d	2024-02-16 18:31:42	Anomaly
02f24444-383a-48e9-b782-b90a5ce3991c	2024-02-16 18:31:42	Anomaly
0a554255-cf0e-4275-8b96-358a4dd76a75	2024-02-16 18:31:42	Anomaly

Figure 5.7: Framework dashboard data analytics - outlier detection - during one of the experiments.

to be actualized each minute and presents results such as “Number of analyzed datasets”, “Number of normal packets analyzed”, and “Number of packages with anomalies”.

Next, we run the internal attacks script using TCPreplay. The internal attack injection script is executed, and the framework analyzes and presents the results. At the end of the injection, the “Number of normal packets” and the “Number of packet anomalies” are displayed, and a new injection is performed. The internal attack injection script was run five times in each experiment. In each experiment, the anomaly detection model was retrained. Each experimental results are presented in Tables 5.3, 5.4, 5.5, 5.6, 5.7.

It injected 35,000 network packets using TCPreplay, and the framework collected approximately 16,000. In the TCPdump, the colletor network packet tool has a filter not to collect broadcast network packets because they can contain duplicate packets. The majority of packets injected into the network interface were correctly identified as anomalies. However, there was a difference between the total number of injected packets, which can be explained by packet losses or errors during data injection by the TCPreplay tool.

In the injection attack, TCPreplay changes the packet header address. This change makes it impossible to compare the dataset because the result of processing the attack data differs from processing the injected attack data. Therefore, the mean and variance were used to understand whether the anomaly detection was carried out correctly. Figure 5.8 presents a chart of each experiment’s average and standard deviation (SD). The average number of packages found by the framework presented a standard deviation of less than 10% in all experiments. Therefore, it is concluded that the anomaly detection model is

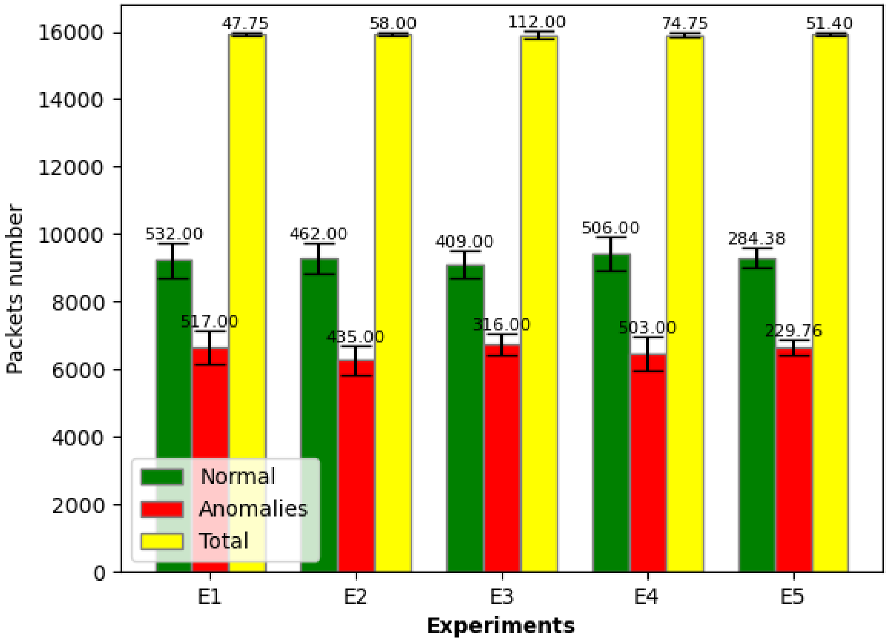


Figure 5.8: Experiments average and standard deviation (SD).

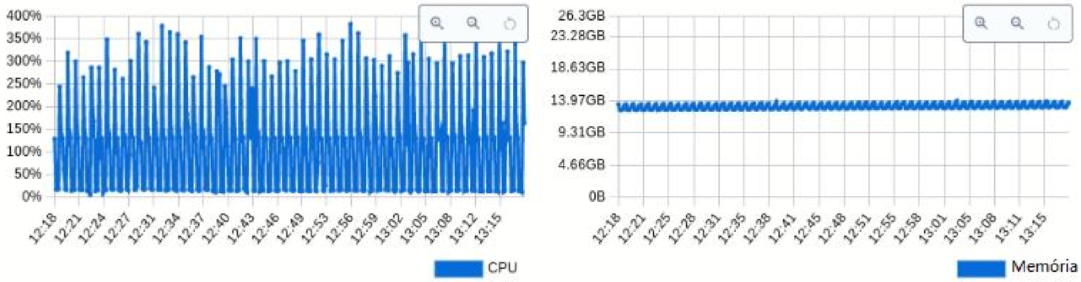


Figure 5.9: CPU and RAM usage during framework execution.

correct with 10% uncertainty.

The packet capture, model execution, and dashboard update occurred within the scheduled time. CPU and RAM metrics were collected by Docker Desktop and are presented in Figure 5.9. The four CPU cores ran between 14% and 380% (each core runs up to 100%). The RAM used was approximately 14 GB. We observe that during the execution model, the CPU percentage is higher. CPU and RAM did not change in the experiments.

Although the framework was developed for 5GCs, according to the evaluation, it can be adjusted for other networks. The initialization time is one hour, which is a point for improvement. In addition, the user with knowledge in ML and container technology is able to change the algorithms and develop new analytics use cases in the framework.

Table 5.3: Results experiment 1.

2*Attack	Number packets		
	Normal	Anomalies	Total
1	10414	5549	15963
2	9395	6609	16004
3	9163	6780	15943
4	9178	6742	15920
5	9230	6646	15876
Average	***9230***	***6646***	***15931***
SD	***532.3753***	***516.8430***	***47.75***

Table 5.4: Results experiment 2.

2*Attack	Number packets		
	Normal	Anomalies	Total
1	9296	6672	15968
2	9221	6603	15824
3	9302	6587	15889
4	9968	5975	15943
5	10226	5717	15943
Average	***9299***	***6281***	***15943***
SD	***461.5536***	***435.1703***	***57.7087***

Table 5.5: Results experiment 3.

2*Attack	Number packets		
	Normal	Anomalies	Total
1	9820	6297	16117
2	9266	6717	15983
3	8958	6901	15859
4	8739	7164	15903
5	9097	6750	15847
Average	***9097***	***6750***	***15903***
SD	***408.5247***	***315.9298***	***111.5132***

Table 5.6: Results experiment 4.

2*Attack	Number packets		
	Normal	Anomalies	Total
1	9561	6248	15809
2	9145	6858	16003
3	9438	6461	15899
4	10190	5762	15952
5	8841	7028	15869
Average	***9438***	***6461***	***15899***
SD	***505.6792***	***503.1220***	***74.7515***

Table 5.7: Results experiment 5.

2***Attack**	**Number packets**		
	Normal	**Anomalies**	**Total**
1	9310	6570	15880
2	9226	6629	15855
3	8824	7114	15938
4	9307	6637	15944
5	9311	6671	15982
Average	***9308.5***	***6633***	***15938***
SD	***284.3847***	***229.7630***	***51.4023***

Table 5.8: Framework anomaly detection model evaluation results.

	Precision	**Recall**	**Accuracy**	**F1-score**	**AUC**
E1	0.99584	0.98866	0.99224	0.99224	0.99225
E2	0.99439	0.98722	0.99079	0.99079	0.99080
E3	0.99605	0.98887	0.99245	0.99245	0.99246
E4	0.99439	0.98784	0.99110	0.99111	0.99111
E5	0.99418	0.98722	0.99069	0.99069	0.99070

5.3 Free5GC and Framework Integration

This section presents the framework and 5GC integration, aiming to show that our framework is integrated with 5GC. This evaluation uses a server computer with RAM 124 GB, HD 1 TB, AMD® Ryzen 9 7950x 16-core processor × 32, and Operational System Ubuntu 20.04.6 LTS 64 bits. The computer server presented has a 5GC.

The Free5GC[3] is an open-source project for 5th-generation mobile core networks. The ultimate goal of this project is to implement the 5GC defined in 3GPP Release 15 (R15) and beyond [26]. Free5GC version presents with UERANSIM[4], the open-source state-of-the-art 5G UE (User Equipment) and RAN (gNodeB) simulator. UE and RAN can be considered a 5G mobile phone and a base station in basic terms. Free5GC and UERANSIM are installed on the server computer presented using container technology.

To emulate the 5GC, we used the Free5GC [25] project version 3.30. It is a GO language open-source project available at Github [11]; it implements the 5GC and 4G Evolved Packet Core (EPC) of 3GPP Release 15. To validate the 5GC functionality, we use the gNB and the User Equipment (UE) provided by the open-source project UERANSIM [31].

In this master's thesis, the dataset [11] used in train models was collected in a

[3] https://free5gc.org

[4] https://github.com/aligungr/UERANSIM

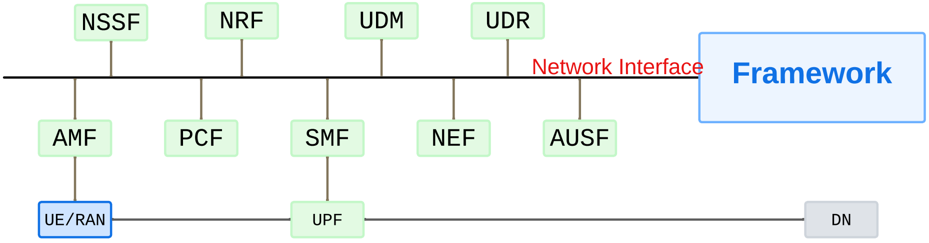


Figure 5.10: 5GC SBA with our framework.

Free5GC[5]. For this reason, Free5GC is chosen. Integration between Free5GC and the framework is possible after adjustments to the Free5GC makefile and docker-compose. Figure 5.10 presents 5GC SBA with our Framework. Our framework is a 5G network function.

It is possible to integrate 5GC and the framework in two ways. The first option involves modifying the framework installation script, auto_run.sh, to be included in the Free5GC Makefile and integrating both into the same docker-compose. The second option involves adding a command in the Free5GC Makefile to install the script auto_run.sh.

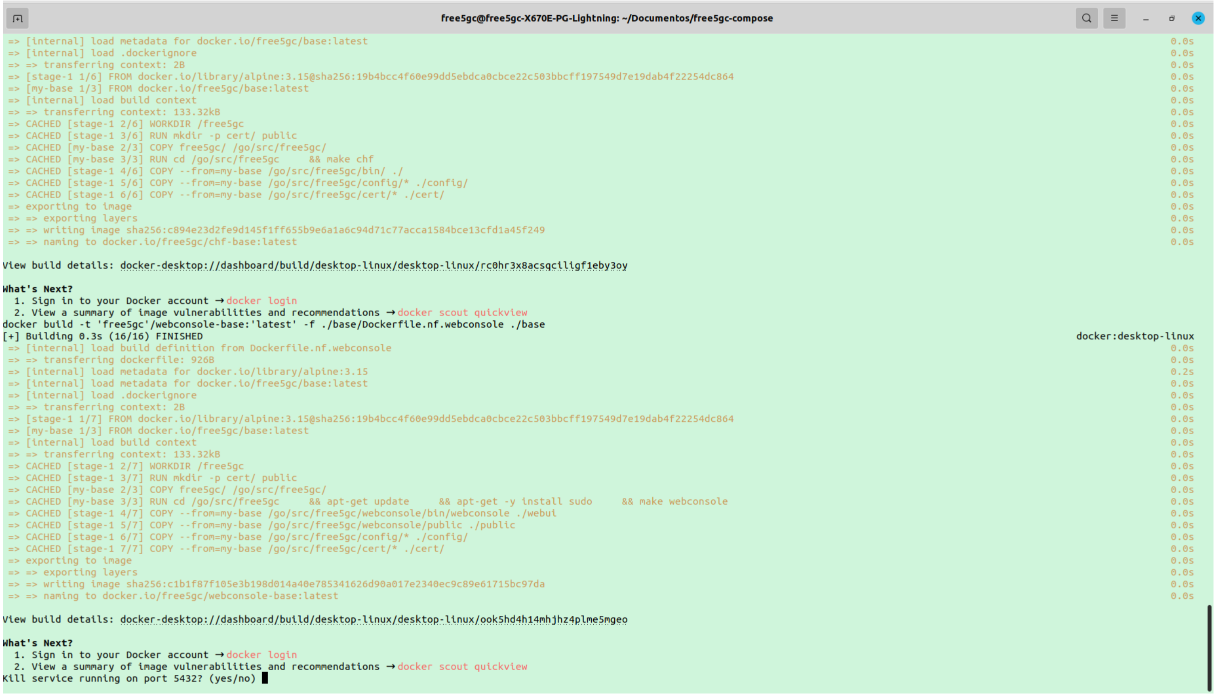
The second option is chosen because it is more conducive to the flexibility and easy installation of the framework. Otherwise, it would be necessary to alter the structures of both the framework and Free5GC, potentially leading to errors and conflicts with our framework objectives.

A command is added to the Free5GC Makefile to execute the framework auto_run.sh script. When a user installs a Free5GC container using the `make` command, both the framework and 5GC are installed, Figure 5.11. To complete the Free5GC installation, the command `docker-compose up -d` is executed. At this point, the framework and Free5GC are integrated.

5.4 Chapter Conclusion

This chapter presented the master thesis evaluation, which consists of three stages: ML algorithms evaluation, framework evaluation, and framework and 5GC integration. The ML algorithms presented an evaluation of some unsupervised and supervised learning algorithms. We used the best ML algorithms to develop an anomaly detection analytics use case in the framework. The evaluation framework used the analytics use case. The

[5] https://free5gc.org


free5gc@free5gc-X670E-PG-Lightning: ~/Documentos/free5gc-compose

```
 => [internal] load metadata for docker.io/free5gc/base:latest                                                0.0s
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [stage-1 1/6] FROM docker.io/library/alpine:3.15@sha256:19b4bcc4f60e99dd5ebdca0cbce22c503bbcff197549d7e19dab4f22254dc864  0.0s
 => [my-base 1/3] FROM docker.io/free5gc/base:latest                                                          0.0s
 => [internal] load build context                                                                             0.0s
 => => transferring context: 133.32kB                                                                         0.0s
 => CACHED [stage-1 2/6] WORKDIR /free5gc                                                                     0.0s
 => CACHED [stage-1 3/6] RUN mkdir -p cert/ public                                                            0.0s
 => CACHED [my-base 2/3] COPY free5gc/ /go/src/free5gc/                                                       0.0s
 => CACHED [my-base 3/3] RUN cd /go/src/free5gc     && make chf                                               0.0s
 => CACHED [stage-1 4/6] COPY --from=my-base /go/src/free5gc/bin/ ./                                          0.0s
 => CACHED [stage-1 5/6] COPY --from=my-base /go/src/free5gc/config/* ./config/                               0.0s
 => CACHED [stage-1 6/6] COPY --from=my-base /go/src/free5gc/cert/* ./cert/                                   0.0s
 => exporting to image                                                                                        0.0s
 => => exporting layers                                                                                       0.0s
 => => writing image sha256:c894e23d2fe9d145f1ff655b9e6a1a6c94d71c77acca1584bce13cfd1a45f249                  0.0s
 => => naming to docker.io/free5gc/chf-base:latest                                                            0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/rc0hr3x8acsqciligf1eby3oy

What's Next?
  1. Sign in to your Docker account → docker login
  2. View a summary of image vulnerabilities and recommendations → docker scout quickview
docker build -t 'free5gc'/webconsole-base:'latest' -f ./base/Dockerfile.nf.webconsole ./base
[+] Building 0.3s (16/16) FINISHED                                                           docker:desktop-linux
 => [internal] load build definition from Dockerfile.nf.webconsole                                            0.0s
 => => transferring dockerfile: 926B                                                                          0.0s
 => [internal] load metadata for docker.io/library/alpine:3.15                                                0.2s
 => [internal] load metadata for docker.io/free5gc/base:latest                                                0.0s
 => [internal] load .dockerignore                                                                             0.0s
 => => transferring context: 2B                                                                               0.0s
 => [stage-1 1/7] FROM docker.io/library/alpine:3.15@sha256:19b4bcc4f60e99dd5ebdca0cbce22c503bbcff197549d7e19dab4f22254dc864  0.0s
 => [my-base 1/3] FROM docker.io/free5gc/base:latest                                                          0.0s
 => [internal] load build context                                                                             0.0s
 => => transferring context: 133.32kB                                                                         0.0s
 => CACHED [stage-1 2/7] WORKDIR /free5gc                                                                     0.0s
 => CACHED [stage-1 3/7] RUN mkdir -p cert/ public                                                            0.0s
 => CACHED [my-base 2/3] COPY free5gc/ /go/src/free5gc/                                                       0.0s
 => CACHED [my-base 3/3] RUN cd /go/src/free5gc     && apt-get update     && apt-get -y install sudo     && make webconsole  0.0s
 => CACHED [stage-1 4/7] COPY --from=my-base /go/src/free5gc/webconsole/bin/webconsole ./webui                0.0s
 => CACHED [stage-1 5/7] COPY --from=my-base /go/src/free5gc/webconsole/public ./public                       0.0s
 => CACHED [stage-1 6/7] COPY --from=my-base /go/src/free5gc/config/* ./config/                               0.0s
 => CACHED [stage-1 7/7] COPY --from=my-base /go/src/free5gc/cert/* ./cert/                                   0.0s
 => exporting to image                                                                                        0.0s
 => => exporting layers                                                                                       0.0s
 => => writing image sha256:c1b1f87f105e3b198d014a40e785341626d90a017e2340ec9c89e61715bc97da                  0.0s
 => => naming to docker.io/free5gc/webconsole-base:latest                                                     0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/ook5hd4h14mhjhz4plme5mgeo

What's Next?
  1. Sign in to your Docker account → docker login
  2. View a summary of image vulnerabilities and recommendations → docker scout quickview
Kill service running on port 5432? (yes/no) █
```

Figure 5.11: Components Free5GC download and framework installation.

framework evaluation section presented experiment steps, metrics, experiment details, and results. In the evaluation, we considered that the model for anomaly detection was correctly trained. The models had F1-score and AUC greater than 98%. We executed five experiments, with five attack injections each, to obtain anomaly and normal packet numbers. We obtained average and standard deviation (SD) packets for each experiment. As a result, the standard deviation represents less than 10% of the average. Therefore, the framework performed correctly to anomaly detection algorithms with a possible error of 10%. We presented the Free5GC and framework integration through container technology.

Chapter 6

Final Conclusion

The NWDAF is a network function designed to collect data from other network functions, generate insights, and facilitate actions that enhance the network's operational efficiency with the support of ML techniques. The development of this network function arose from the need for data analytics and the automation of actions for the operation of 5G networks and beyond. However, according to 3GPP, its implementation needs API integration and 5GC knowledge. Furthermore, we presented several gaps in the existing 5G network data analytics literature. The first gap is a lack of a 5G data analytics framework that is flexible enough to store different analytics use cases, specifically one that allows the inclusion of new analytics algorithms from various analytics use cases. Moreover, according to 3GPP, there is a scarcity of 5G network data analytics frameworks that use open-source software in their architecture. These gaps underscore the necessity for the research presented in this master's thesis and highlight the potential for advancements in this field. Therefore, we proposed a 5G data analytics framework that aims to be easy to install, compliant with 3GPP, and flexible for altering use case algorithms.

We conceived a framework that enables new algorithms due to the architecture design and its components chosen for high data volume. The framework architecture has a user-friendly interface for developing and executing analytics. After the analytics, the framework exposes the result in its dashboard. Its robust storage system can receive and send large data volumes when necessary. We use open-source components to implement the data analytics framework due to access to the source code, freedom of redistribution, collaborative development, open licensing, transparency, active community involvement, interoperability, reduced cost, customization, and enhanced security. Furthermore, each component was chosen according to 3GPP NWDAF functionality to guarantee the framework's operability in the 5GC. Thus, we chose Apache Hadoop, Tcpdump, Apache Zeppelin, Apache Flink, PostgreSQL, and Grafana as the framework architecture components. Next, we implemented the designed architecture. Lastly, we evaluated the framework using an anomaly detection analytics use case, a 5G cybersecurity dataset, and an ML algorithm.

We performed evaluations in three steps. First, we presented ML algorithm evaluation results, also available at [53] [69]. Second, five experiments validated the framework functionalities in the anomaly detection use case. Next, we integrated the framework and 5GC. Free5GC was chosen because it was the core used by the authors in [11] when creating the dataset used in training the ML algorithms. Among the supervised learning algorithms, the Bagging regressor had the best evaluation. Among the unsupervised learning algorithms, the Elliptic Envelope had the best evaluation. We used the best algorithms to input the analytics use case in the framework to benefit from the learning acquired. As a result, the anomaly detection model presented an F1-score and AUC of more than 98%, and the framework evaluation showed the correct operation of the framework for the anomaly detection use case with less than 10% uncertainty obtained in standard deviation. The framework and Free5GC were correctly integrated using container technology. In addition, the framework was installed, and data was collected correctly, making it easy to install.

The conclusion of this work not only marks the end of an academic stage and represents a period of intense personal and professional growth. I developed fundamental technical and interpersonal skills for my career during the master‘s thesis preparation. I learned essential computing subjects such as algorithms, data structures, programming, theory of computation, and software engineering. Also, I reinforced my previous knowledge of operating systems, wireless networks, computer networks, and cellular networks. Designing the framework allowed me to integrate the knowledge acquired in my telecommunications engineering education with the new learnings from the master's program, such as ML, big data, software engineering, and cybersecurity. Furthermore, I developed scientific research skills and participated in projects and group work. The combination of theoretical and practical knowledge acquired throughout the course enriched my education and equipped me with the necessary tools to face complex professional challenges.

Future works for this master's thesis encompass several directions for improving the 5G network data analytics framework. Initially, it is essential to incorporate additional use cases, thereby expanding its applicability. Furthermore, exploring new use cases is crucial to ensure the framework's relevance across different scenarios. Integrating the framework into various 5GC represents another critical research direction, aiming to guarantee its interoperability and efficacy in diverse environments. Adding functionality for algorithms operating simultaneously will enable more comprehensive and efficient analytics of the entire network across multiple use cases. The framework evaluation to support centralized, distributed, and hybrid NWDAF operations is necessary to adapt to different network architectures and optimize computational efficiency. It is intended to generate proprietary data to enhance the precision of evaluations and make these data available to the research community. In future work, the automation of network management actions will be incorporated to improve the 5G network. Also, deepening the evaluation of the framework under high 5G network traffic rates is crucial to verify its scalability and performance under realistic conditions. Finally, studying unsupervised algorithms applied to the dataset used in this work will allow for a better understanding of the unsatisfactory results observed during the framework's evaluation and improve knowledge in this class of algorithms. These future works represent significant contributions to the continuous development and refinement of the framework.

Bibliography

[1] 3GPP. Ai/ml management for 5g systems, 2023. [Online] Available: https://www.3gpp.org/technologies/ai-ml-management. Last accessed on: September, 2023.

[2] 3rd Generation Partnership Project. 5g; 5g security assurance specification (scas); network data analytics function (nwdaf) (3gpp ts 33.521 version 18.0.0 release 18), 2023. [Online] Available: https://www.3gpp.org/ftp/Specs/archive/33_series/33.521/33521-i00.zip. Last accessed on: April, 2024.

[3] 3rd Generation Partnership Project. 5g; architecture enhancements for 5g system (5gs) to support network data analytics services (3gpp ts 23.288 version 18.3.0 release 18), 2023. [Online] Available: https://www.3gpp.org/ftp/Specs/archive/23_series/23.288/23288-i50.zip. Last accessed on: April, 2024.

[4] 3rd Generation Partnership Project. Technical report 23.501 - "technical specification group services and system aspects; system architecture for the 5g system (5gs);", v18.4.0, dec 2023., 2023. [Online] Available: www.3gpp.org/ftp/Specs/archive/23_series/23.501/23501-i40.zip. Last accessed on: March, 2024.

[5] 3rd Generation Partnership Project. About 3gpp, 2024. [Online] Available: https://www.3gpp.org/about-us. Last accessed on: March, 2024.

[6] 3rd Generation Partnership Project (3GPP) TS 29.520. "5g system; network data analytics services, version 18.3.0 release 18", 2023. [Online] Available: https://www.3gpp.org/ftp/Specs/archive/29_series/29.520/29520-i30.zip. Last accessed on: October, 2023.

[7] Khizar Abbas, Talha Ahmed Khan, Muhammad Afaq, Javier Jose Diaz Rivera, and Wang-Cheol Song. Network data analytics function for ibn-based network slice lifecycle management. In *2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 148–153, 2021.

[8] Khizar Abbas, Talha Ahmed Khan, Muhammad Afaq, and Wang-Cheol Song. Ensemble learning-based network data analytics for network slice orchestration and management: An intent-based networking mechanism. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–5, 2022.

[9] Mohiuddin Ahmed and Al-Sakib Khan Pathan. *Data analytics: concepts, techniques, and applications*. Crc Press, 2018.

[10] Abdulaziz Aldoseri, Khalifa N. Al-Khalifa, and Abdel Magid Hamouda. Re-thinking data strategy and integration for artificial intelligence: Concepts, opportunities, and challenges. *Applied Sciences*, 13(12), 2023.

[11] Matthew W Anderson, Denver S Conger, Brendan G Jacobson, Damon R Spencer, Edward J Goodell, Bryton J Petersen, Cooper W Coldwell, Matthew R Sgambati, Safety USDOE Office of Environment, Health, and Security. Simulated 5g network traffic dataset, 8 2022.

[12] Erik Aumayr, Giuseppe Caso, Anne-Marie Bosneag, Almudena Diaz Zayas, Özgü Alay, Bruno Garcia, Konstantinos Kousias, Anna Brünstrom, Pedro Merino Gomez, and Harilaos Koumaras. Service-based analytics for 5g open experimentation platforms. *Computer Networks*, 205:108740, 2022.

[13] Manish Bhavsar, Pradeep Deshmukh, and Khushal Khairnar. Machine learning and 5g charging function with network analytics function for network slice as a service. In *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, pages 444–448, 2022.

[14] Amine Boukhtouta, Taous Madi, Makan Pourzandi, and Hyame Alameddine A. Cloud native applications profiling using a graph neural networks approach. In *2022 IEEE Future Networks World Forum (FNWF)*, pages 220–227, 2022.

[15] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[16] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, may 2000.

[17] Kuan–Hsiang Chen and Huai–Sheng Huang. Meta-nwdaf: A meta-learning based network data analytic function for internet traffic prediction. In *2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 01–04, 2022.

[18] Ali Chouman, Dimitrios Michael Manias, and Abdallah Shami. Towards supporting intelligence in 5g/6g core networks: Nwdaf implementation and initial analysis. In *2022 International Wireless Communications and Mobile Computing (IWCMC)*, pages 324–329, 2022.

[19] Cooper Coldwell, Denver Conger, Edward Goodell, Brendan Jacobson, Bryton Petersen, Damon Spencer, Matthew Anderson, and Matthew Sgambati. Machine learning 5g attack detection in programmable logic. In *2022 IEEE Globecom Workshops (GC Wkshps)*, pages 1365–1370, 2022.

[20] Instituto Nacional de Pesquisas Espaciais – INPE. Laboratório Associado de Sensores e Materiais LAS/CTE/INPE. Experimento célula solar do satélite scd2, 2021. [Online] Available: http://www.las.inpe.br/~veissid/por20.html. Last accessed on: December, 2023.

[21] Hale Donertasli and Madhukiran Medithe. Nwdaf udi (use-case development interface) for end-to-end ai enabled 5g and beyond networks. In *2022 International Conference on Artificial Intelligence of Things (ICAIoT)*, pages 1–6, 2022.

[22] Ericsson. Ericsson mobility report, 2021. [Online] Available https://www.ericsson.com/en/mobility-report/reports/june-2021. Last accessed on: november 7, 2023.

[23] Rui Ferreira, João Fonseca, João Silva, Mayuri Tendulkar, Paulo Duarte, Marco Araújo, Raul Barbosa, Bruno Mendes, and Adriano Goes. Demo: Enhancing network performance based on 5g network function and slice load analysis. In *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 340–342, 2023.

[24] Apache Flink. Apache flink, 2023. [Online] Available: https://flink.apache.org. Last accessed on: October, 2023.

[25] Free5GC, 2023. [Online] Available: https://github.com/free5gc/free5gc. Last accessed on: October, 2023.

[26] Free5GC, 2024. [Online] Available: https://free5gc.org. Last accessed on: March, 2024.

[27] Jerome H Friedman, Forest Baskett, and Leonard J Shustek. An algorithm for finding nearest neighbors. *IEEE Transactions on computers*, 100(10):1000–1006, 1975.

[28] Erich Gamma. *Padrões de projetos: soluções reutilizáveis*. Bookman editora, 2009.

[29] Amitabha Ghosh, Andreas Maeder, Matthew Baker, and Devaki Chandramouli. 5g evolution: A view on 5g cellular technology beyond 3gpp release 15. *IEEE Access*, 7:127639–127651, 2019.

[30] Anastasios Giannopoulos, Sotirios Spantideas, Nikolaos Kapsalis, Panagiotis Gkonis, Lambros Sarakis, Christos Capsalis, Massimo Vecchio, and Panagiotis Trakadas. Supporting intelligence in disaggregated open radio access networks: Architectural principles, ai/ml workflow, and use cases. *IEEE Access*, 10:39580–39595, 2022.

[31] Ali Gung. Ueransim, 2023. [Online] Available: https://github.com/aligungr/UERANSIM. Last accessed on: October, 2023.

[32] Apache Hadoop. Apache hadoop, 2023. [Online] Available: https://hadoop.apache.org. Last accessed on: September, 2023.

[33] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

[34] S. Hettich and S. Bay. The uci kdd archive, 2005. [Online] Available: http://kdd.ics.uci.edu. Last accessed on: March, 2024.

[35] Mohammad Arif Hossain, Abdullah Ridwan Hossain, Weiqi Liu, Nirwan Ansari, Abbas Kiani, and Tony Saboorian. A distributed collaborative learning approach in 5g+ core networks. *IEEE Network*, pages 1–8, 2023.

[36] Sara Khalil Mohamed Ibrahim. *Spacecraft Performance Analysis and Fault Diagnosis Using Telemetry-mining*. PhD thesis, Doctoral dissertation, Faculty of Engineering, Zagazig University, Egypt, 2018.

[37] Shun ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4):185–196, 1993.

[38] Mats Johansson J. Network slicing: Top 10 use cases to target, stockholm, sweden, 2021, 2021. [Online] Available: https://www.ericsson.com/en/blog/2021/6/network-slicing-top-10-industries. Last accessed on: November, 2023.

[39] Youbin Jeon, Hyeonjae Jeong, Sangwon Seo, Taeyun Kim, Haneul Ko, and Sangheon Pack. A distributed nwdaf architecture for federated learning in 5g. In *2022 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–2, 2022.

[40] Jaeseong Jeong, Dinand Roeland, Jesper Derehag, Åke Ai Johansson, Venkatesh Umaashankar, Gordon Sun, and Göran Eriksson. Mobility prediction for 5g core networks. *IEEE Communications Standards Magazine*, 5(1):56–61, 2021.

[41] Taeyun Kim, Joonwoo Kim, Haneul Ko, Sangwon Seo, Youbin Jcon, Hyeonjae Jeong, Seunghyun Lee, and Sangheon Pack. An implementation study of network data analytic function in 5g. In *2022 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–3, 2022.

[42] Grafana Labs. Grafana documentation, 2018. [Online] Available: https://grafana.com/docs/. Last accessed on: March, 2024.

[43] Seunghyun Lee, Jaewook Lee, Taeyun Kim, Daeyoung Jung, Inho Cha, Dongju Cha, Haneul Ko, and Sangheon Pack. Design and implementation of network data analytics function in 5g. In *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 757–759, 2022.

[44] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008.

[45] Jun Liu, Feng Liu, and Nirwan Ansari. Monitoring and analyzing big traffic data of a large-scale cellular network with hadoop. *IEEE Network*, 28(4):32–39, 2014.

[46] Tomasz Łuczak and Boris Pittel. Components of random forests. *Combinatorics, Probability and Computing*, 1(1):35–52, 1992.

[47] Ciro Macedo, Hudson Romualdo, Cristiano Both, Antonio Oliveira-Jr, and Kleber Cardoso. Nwdaf habilitando inteligência artificial em operações de busca e salvamento assistidas por vants. In *Anais do I Workshop de Redes 6G*, pages 13–18, Porto Alegre, RS, Brasil, 2021. SBC.

[48] Dimitrios Michael Manias, Ali Chouman, and Abdallah Shami. An nwdaf approach to 5g core network signaling traffic: Analysis and characterization. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pages 6001–6006, 2022.

[49] Henrique D. Moura, Júnia Maísa Oliveira, Daniel Soares, Daniel F. Macedo, and Marcos A. M. Vieira. Improved video qoe in wireless networks using deep reinforcement learning. In *2023 19th International Conference on Network and Service Management (CNSM)*, pages 1–7, 2023.

[50] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[51] Júnia Oliveira, Emanuela Ferraz, Vinícius Oliveira, Daniel Macedo, and José Nogueira. Pipa: Uma solução integradora de políticas de controle de acesso a recursos e de gerenciamento de identidades. In *Anais Estendidos do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 64–71, Porto Alegre, RS, Brasil, 2023. SBC.

[52] Júnia Oliveira, Vinícius Oliveira, Daniel Macedo, Dorgival Guedes, and José Nogueira. Abordagem confiança zero aplicada a ambientes computacionais big data: um estudo de caso. In *Anais do XXVII Workshop de Gerência e Operação de Redes e Serviços*, pages 127–140, Porto Alegre, RS, Brasil, 2022. SBC.

[53] Júnia Maísa Oliveira, Jônatan Almeida, Daniel Macedo, and José Marcos Nogueira. Comparative analysis of unsupervised machine learning algorithms for anomaly detection in network data. In *2023 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6, 2023.

[54] Júnia Maísa Oliveira, Marcos Carvalho, Daniel Macedo, Cassio G. Rego, and José Marcos Nogueira. Data consumption and user experience in video lecture sessions via mobile telephony network. In *2023 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6, 2023.

[55] Sankar K Pal and Sushmita Mitra. Multilayer perceptron, fuzzy sets, classifiaction, 1992.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[57] Ramon Perez, Jaime Garcia-Reinoso, Aitor Zabala, Pablo Serrano, and Albert Banchs. A monitoring framework for multi-site 5g platforms. In *2020 European Conference on Networks and Communications (EuCNC)*, pages 52–56, 2020.

[58] Tan Pn, M Steinbach, and V Kumar. *Introduction to data mining*. Addison-Wesley, 2005.

[59] Postgresql. Postgresql, 2023. [Online] Available: https://www.postgresql.org. Last accessed on: September, 2023.

[60] Madanagopal Ramachandran, T. Archana, V. Deepika, A. Arjun Kumar, and Krishna M. Sivalingam. 5g network management system with machine learning based analytics. *IEEE Access*, 10:73610–73622, 2022.

[61] Peter J. Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

[62] Georgios Samaras, Vasileios Theodorou, Dimitris Laskaratos, Nikolaos Psaromanolakis, Marinela Mertiri, and Alexandros Valantasis. Qmp: A cloud-native mlops automation platform for zero-touch service assurance in 5g systems. In *2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pages 86–89, 2022.

[63] Susanna Schwarzmann, Clarissa Cassales Marquezan, Riccardo Trivisonno, Shinichi Nakajima, Vincent Barriac, and Thomas Zinner. Ml-based qoe estimation in 5g networks using different regression techniques. *IEEE Transactions on Network and Service Management*, 19(3):3516–3532, 2022.

[64] Muhammad Shafiq, Xiangzhan Yu, Asif Ali Laghari, Lu Yao, Nabin Kumar Karn, and Foudil Abdessamia. Network traffic classification techniques and comparative

analysis using machine learning algorithms. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 2451–2455, 2016.

[65] Alain Sultan. 5g system overview, 2022. [Online] Available: https://www.3gpp.org/technologies/5g-system-overview. Last accessed on: December, 2023.

[66] Alain Sultan. 5g system overview, 2022. [Online] Available: https://www.3gpp.org/technologies/5g-system-overview. Last accessed on: September, 2023.

[67] Philip H Swain and Hans Hauska. The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, 15(3):142–147, 1977.

[68] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. A detailed analysis of the KDD CUP 99 data set. In *IEEE Symposium on Comp. Intelligence for Security and Defense Applications*, 2009.

[69] Isabela Tavares, Júnia Maísa Oliveira, André Ferreira Teixeira, Marconi de Arruda Pereira, Marcos Tomio Kakitani, and José Marcos Nogueira. Machine learning algorithms applied to telemetry data of scd-2 brazilian satellite. In *Proceedings of the 2022 Latin America Networking Conference*, LANC '22, page 50–58, New York, NY, USA, 2022. Association for Computing Machinery.

[70] Positive Technologies. 5g standalone core security research, 2020. [Online] Available: https://img.lightreading.com/5gexchange/downloads/5G-Standalone-core-security-research.pdf. Last accessed on: October, 2023.

[71] Positive Technologies. Threat vector: Gtp, 2020. [Online] Available: https://img.lightreading.com/5gexchange/downloads/5G-Standalone-core-security-research.pdf. Last accessed on: October, 2023.

[72] R Vidhya, P Karthik, and Satish Jamadagni. Anticipatory qoe mechanisms for 5g data analytics. In *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*, pages 523–526, 2020.

[73] Sami Virpioja, Ville T Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, 52(2):45–90, 2011.

[74] Dan Wang, Yongjing Li, Aihua Li, Xiaonan Sh, and Wei Wang. 5g-advanced technology evolution from a network perspective 2.0, 2022. [Online] Available: https://www-file.huawei.com/-/media/corporate/pdf/news/5g-advanced%20technology%20evolution%20from%20a%20network%20perspective(2022).pdf?la=en. Last accessed on: April, 2023.

[75] Min Xie, Foivos Michelinakis, Thomas Dreibholz, Joan S. Pujol-Roig, Sara Malacarne, Sayantini Majumdar, Wint Yi Poe, and Ahmed M. Elmokashfi. An exposed closed-loop model for customer-driven service assurance automation. In *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, pages 419–424, 2021.

[76] Apache Zeppelin. Apache zeppelin, 2023. [Online] Available: https://zeppelin.apache.org. Last accessed on: September, 2023.