DOUGLAS CHAGAS DA SILVA

An Approach to Network Slice Selection Function for Multi-domain Environments on 5G Mobile Networks and Future Communication Systems

São Paulo
2023

DOUGLAS CHAGAS DA SILVA

An Approach to Network Slice Selection Function for Multi-domain Environments on 5G Mobile Networks and Future Communication Systems

Doctoral thesis presented to Escola Politécnica of Universidade de São Paulo (USP) to obtain the degree of Doctor of Science.

São Paulo
2023

DOUGLAS CHAGAS DA SILVA

An Approach to Network Slice Selection Function for Multi-domain Environments on 5G Mobile Networks and Future Communication Systems

Corrected Version

Doctoral thesis presented to Escola Politécnica of Universidade de São Paulo (USP) to obtain the degree of Doctor of Science.

Concentration area:

Computer Engineering

Advisor:

Dr. Regina Melo Silveira

São Paulo
2023

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo - SP, 28 de Novembro de 2023.

ASSINADO DIGITALMENTE
DOUGLAS CHAGAS DA SILVA
A conformidade com a assinatura pode ser verificada em:
http://serpro.gov.br/assinador-digital
SERPRO

Assinatura do autor: ____________________________

Documento assinado digitalmente
gov.br
REGINA MELO SILVEIRA
Data: 07/12/2023 10:21:06-0300
Verifique em https://validar.iti.gov.br

Assinatura do orientador:_

Catalogação-na-publicação

DA SILVA, DOUGLAS CHAGAS
An Approach to Network Slice Selection Function for Multi-domain Environments on 5G Mobile Networks and Future Communication Systems / D. C. DA SILVA -- versão corr. -- São Paulo, 2023.
172 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1.Network Slice Selection Function 2.Multi-criteria Decision Methods 3.Machine learning 4.Networks Softwarization 5.5G I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

ACKNOWLEDGMENTS

First and foremost, I would like to extend my heartfelt gratitude to God for the precious gifts of life, health, and countless opportunities bestowed upon me.

To my parents, João (in memoriam) and Josefa, I owe an eternal debt of gratitude for being unwavering pillars of support, guiding me with their exemplary character in every endeavor I have embarked upon in my life.

Moreover, I am profoundly thankful to my dear grandmother, Maria Rosa, whose wise guidance and presence in my life have been an invaluable source of inspiration.

Even during the most challenging times, my beloved daughter, Maria Luíza, even being just a little girl has been a constant source of strength, motivating me to persevere.

I am indebted to my advisor, Prof. Regina Silveira, for her consistent and enriching support throughout my doctoral journey, which has played a crucial role in shaping my academic growth.

Additionally, my heartfelt appreciation goes out to the professors of PPGEE/USP and PCS/POLI/USP, especially Prof. Graça Bressan, and Prof. Alexandre Patriota of IME/USP, for their unwavering dedication to their profession and their ever-readiness to lend a helping hand to anyone in need.

Furthermore, I am immensely grateful to the friends I made at LARC/USP (in alphabetical order), Armando Menéndez, Elaine Freitas, Fernando López, José Carlos Menéndez, José Olímpio, Loubrys Reinoso, Patrícia Reina, and Wagner Oliveira. Our meaningful connections, fruitful discussions, and technical contributions have enriched my journey in immeasurable ways.

I cannot overlook the invaluable support from the entire GPRISA/UNITINS team, and especially my friends and colleagues, Marco Firmino and Luís Gustavo Rodriguez. Our constant exchange of information and collaboration have been truly transformative.

Lastly, I extend my deepest thanks to UNITINS and UFT for their generous financial support, without which this work would not have been possible.

Thank you all for all your support!

"You step onto the road, and if you don't keep your feet, there is no knowing where you might be swept off to."

The Lord of the Rings
J.R.R. Tolkien

RESUMO

Uma abordagem para o serviço de Seleção de Slices para Ambientes Multidomínio em Redes Móveis 5G e Sistemas de Comunicação Futuros

O serviço Network Slice Selection Function (NSSF) em ambientes de tecnologia heterogêneos é um problema complexo, que ainda não tem uma solução totalmente aceitável. Assim, a implementação de novas estratégias de seleção de slices representa uma questão importante em desenvolvimento, principalmente devido à crescente demanda em aplicações e cenários envolvendo redes 5G e futuras. Este trabalho apresenta uma solução integrada para o problema NSSF, denominado Network Slice Selection Function Decision-Aid Framework (NSSF DAF), que consiste em uma solução distribuída, onde uma parte é executada no equipamento do usuário (e.g. smartphones, VANTs, Brokers IoT), funcionando como um serviço transparente e outra à borda da operadora ou do provedor de serviços. Para tanto, protocolos e ferramentas de software são usados para classificar os *slices*. Neste trabalho, 14 métodos multicritérios são empregados para auxiliar na tomada de decisão, sendo eles: ARAS, COCOSO, CODAS, COPRAS, EDAS, MABAC, MAIRCA, MARCOS, MOORA, OCRA, PROMETHEE II, SPOTIS, TOPSIS e VIKOR. O objetivo geral consiste em verificar a semelhança entre esses métodos e aplicações no processo de classificação e seleção de slices, considerando um cenário específico. Para realizar a seleção é utilizado aprendizado de máquina por meio do algoritmo de agrupamento K-means, adotando uma solução híbrida para a implementação e operação do serviço NSSF em ambientes de slices multi-domínio em redes móveis heterogêneas. Testes de bancada foram conduzidos visando validar a abordagem proposta, mapeando e correlacionando os requisitos dos serviços com os *slices* disponíveis. Os resultados indicam uma possibilidade real de oferecer uma solução completa para o problema NSSF que pode ser implementada na borda ou no núcleo da rede, ou mesmo na própria Estação Rádio Base 5G, sem custo computacional incremental ao equipamento do usuário final, garantindo a qualidade de experiência adequada ao consumo dos seus serviços.

Palavras-chave: Seleção de Slices. Decisão Multicritério. Aprendizado de Máquina. Softwarização de Rede. Redes 5G.

ABSTRACT

The Network Slice Selection Function (NSSF) in heterogeneous technology environments is a complex problem, which still does not have a fully acceptable solution. Thus, the implementation of new network selection strategies represents an important issue in development, mainly due to the growing demand for applications and scenarios involving 5G and future networks. This work then presents an integrated solution for the NSSF problem, called Network Slice Selection Function Decision-Aid Framework (NSSF DAF), which consists of a distributed solution in which a part is executed on the user's equipment (e.g. smartphones, Unmanned Aerial Vehicles, IoT brokers), functioning as a transparent service, and another at the Edge of the operator or service provider. It requires low consumption of computing resources from mobile devices and offers complete independence from the network operator. For this purpose, protocols and software tools are used to classify slices. This work employs fourteen multicriteria methods to aid decision-making: ARAS, COCOSO, CODAS, COPRAS, EDAS, MABAC, MAIRCA, MARCOS, MOORA, OCRA, PROMETHEE II, SPOTIS, TOPSIS and VIKOR. The general objective is to verify the similarity among these methods and applications in the slice classification and selection process, considering a specific scenario, towards the framework. It also uses machine learning through the K-means clustering algorithm, adopting a hybrid solution to implement and operate the NSSF service in multi-domain slicing environments of heterogeneous mobile networks. Testbeds were conducted to validate the proposed framework, mapping the adequate quality of service requirements. The results indicate a real possibility of offering a complete solution to the NSSF problem that can be implemented in Edge, in Core, or even in 5G Radio Base Station itself, without the incremental computational cost of the end user's equipment, allowing the adequate quality of experience.

Keywords: Network Slice Selection Function. Multi-criteria Decision Methods. Machine learning. Networks Softwarization. 5G.

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

3GPP	3rd Generation Partnership Project
5G	Fifth Generation Mobile Networks
5GC	5G Core
ABC	Always Best Connected
AF	Application Function
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
AIOps	Artificial Intelligence for IT Operations
AMF	Access and Mobility Management Function
AP	Access Point
API	Application Programming Interface
AR	Augmented Reality
AUSF	Authentication Server Function
BAM	Bayesian Attractor Model
CBR	Constant Bit Rate
CD	Continuous Delivery
CI	Continuous Integration
CLI	Command Line Interface
CNF	Cloud-Native Network Functions
CNN	Convolutional Neural Networks
COM	Center of Maximum
CPE	Customer Premise Equipment
CU	Centralized Unit
CUPS	Control User Plane Split
DevOps	Software Development and Operations
DPDK	Data Plane Development Kit
DPI	Deep Packet Inspection
DPMM	Dirichlet Process Mixture Model

DU	Distributed Unit
E2E	End-to-End
EMCO	Edge Multi-Cluster Orchestrator
EPoP	Edge Points of Presence
ETSI	European Telecommunications Standards Institute
EVPN	VPN Ethernet
FAHP	Fuzzy Analytic Hierarchy Process
GA	Genetic Algorithm
GNS3	Graphical Network Simulator 3
GRA	Grey Relation Analysis
GTP	GPRS Tunneling Protocol
GUI	Graphical User Interface
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force
IoT	Internet of Things
IoV	Internet of Vehicles
IQR	Interquartile Range
ISP	Internet Service Provider
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LSTM	Long-Short Term Memory
MCDA	Multiple-criteria decision analysis
MCDM	Multiple-criteria decision-making
MEC	Multi-Access Edge Computing
MEW	Multiplicative Exponential Weight
ML	Machine Learning
MLOps	Machine Learning Operations
MM	Mobility Management
MPLS	Multi-Protocol Label Switching
MQTT	Message Queuing Telemetry Transport
NEF	Network Exposure Function

NF	Network Functions
NFV	Network Functions Virtualization
NGAP	NG Application Protocol
NGMN	Next Generation Mobile Networks
NR	New Radio
NRF	Network Repository Function
NS	Network Slicing
NSA	Non-Standalone Architecture
NSSF DAF	Network Slice Selection Function Decision-Aid Framework
NSSF	Network Slice Selection Function
O-RAN	Open Radio Access Network
ONAP	Open Network Automation Platform
ONF	Open Networking Foundation
OPEX	Operating Resources
OSM	Open Source Mano
PCF	Policy Control Function
QoE	Quality of Experience
QoS	Quality of Service
QoV	Quality of Video
RAN	Radio Access Network
RAT	Radio Access Technologies
RQ	Research Questions
RSS	Received Signal Strength
RTT	Round-Trip Time
SA	Standalone Architecture
SAW	Simple Additive Weighting
SCTP	Stream Control Transmission Protocol
SDN	Software-Defined Networking
SLA	Service Layer Agreement
SMART	Simple Multiattribute Rating Technique
SMF	Session Management Function

SMS	Short Messaging Service
SR	Segment Router
SRQ:	Secondary Research Questions
SSE	Root Mean Square Error
UAV	Unmanned Aerial Vehicle
UDM	Unified Data Management
UE	User Equipment
UPF	User Plane Function
V2I	Vehicle-to-Infrastructure
V2N	Vehicle-to-Network
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VNF	Virtual Network Function
VoD	Video on Demand
VR	Virtual Reality
VXLAN	Virtual Extensible Lan
WSN	Wireless Sensor Networks
XML	Extensible Markup Language
YAML	YAML Ain't Markup Language

CONTENTS

1 INTRODUCTION

This introductory chapter presents the reasons, research questions, objectives, and background for this work. It has been organized into the five following sections: Section 1.1 gives relevant background information; Section 1.2 addresses the research questions and objective; Section 1.3 looks at the main reasons for this study; Section 1.4 describes the main assumptions used by Network Slice Selection Function Decision-Aid Framework (NSSF DAF) designing, and testing the slice selection service framework; and Section 1.5 provides an overview of the structure of this document.

1.1 Background

Convergence among networks of different technologies has become a reality. The processing power of mobile devices and the diversity of services that can be used with them shape the way in which access technology infrastructures are modeled. The term convergence hence refers to the possibility of providing user access through different technologies, enabling the use of various services such as voice, video, and data in general (SILVA et al., 2022b),(MORGADO et al., 2018).

In applying the concept of the Network Slicing (NS), an avenue to new mobile network solutions can be provided, including 5G networks (Fifth Generation Mobile Networks) and future communication systems, as a result of the stricter requirements in terms of their Key Performance Indicators (KPIs) — isolation, latency, mobility, peak data rate, among others. The slice constitutes a virtual segment of the network that aims at offering specific services and providing isolation between the applications. Although several proposals have pointed out paths in domains that involve heterogeneous technologies (*e.g.* WLAN, WiMAX, UMTS, HSPA, LTE and IMT-2020), it is not possible to aggregate several functionalities in a single and fully functional approach so as to set the operation and management mechanisms of each slice, or to provide subsidies for scalability, orchestration, and support for the decision making process (MORGADO et al., 2018; YI et al., 2018).

The task of selecting a Radio Access Network (RAN) in an environment of heterogeneous technologies is difficult, since operators can provide specific types of slices directly in order to meet the requirements of an application or even multiple slices for the requirements of the same user. Therefore, there is no solution or technique whatsoever that understands all the aspects and mechanisms of access to these technologies (KIM; KIM, 2019; YOU et al., 2019). The implementation of new selection techniques becomes necessary due to the demand in the growing use of vehicular networks, patient monitoring, smart cities, and Internet of Things (IoT), among other technologies and scenarios involving network convergence, mobility management, and service continuity in 5G networks and beyond (BARAKABITZE et al., 2020; CHAHAL; HARIT, 2019; WEI; ZHANG; FAN, 2018; BOJKOVIC; BAKMAZ; BAKMAZ, 2019; ÇATAK; DURAK-ATA, 2016; AUMAYR et al., 2022; WU et al., 2022).

In this context, a study of the use of computational tools for classifying slices was carried out, with the objective to support the decision-making process using aids for our proposed NSSF DAF. Fourteen strategies were suggested, namely Aras (Additive Ratio Assessment), Cocoso (Combined Compromise Solution), Codas (Combinative Distance-based Assessment), Copras (Complex Proportional Assessment), Edas (Evaluation based on Distance from Average Solution), Mabac (Multiattributive Border Approximation Area Comparison), Mairca (Multi-Attributive Ideal–Real Comparative Analysis), Marcos (Measurement of Alternatives and Ranking according to Compromise Solution), Moora (Multi-Objective Optimization on the basis of Ratio Analysis), Ocra (Operational Competitiveness Rating), Promethee II (Preference Ranking Organization Method for Enrichment Evaluations), Spotis (Stable Preference Ordering Towards Ideal Solution), Topsis (Technique for Order Preference by Similarity to Ideal Solution), and Vikor (Visekriterijumska Optimizacija i Kompromisno Resenje)(SILVA et al., 2022b),(ALINEZHAD; KHALILI, 2019), (HEZER; GELMEZ; ÖZCEYLAN, 2021), (WANG et al., 2020),(ULUTAş et al., 2020),(KUNDAKCı, 2019),(NGUYEN et al., 2022a),(DEZERT et al., 2020).

The general purpose was, first, to verify the similarity among these methods and applications in the slice classification and selection processes, considering a range of different scenarios, and then to propose a framework that provides the user with the best experience. Machine Learning (ML) with the K-means clustering algorithm as well as Fuzzy Logic were selected, adopting a hybrid solution to the deployment and operation of the Network Slice Selection Function (NSSF) service in multi-domain slicing environments of heterogeneous mobile networks. To validate the framework, testbeds were conducted to the mapping of the necessary Quality of Service (QoS) requirements in order to guarantee the established SLAs (Service Level Agreement).

Thus, there is a promising solution for 5G mobile networks and beyond, which is a new approach that employs techniques aimed at the integration and interoperability between RAN or Open Radio Access Network (O-RAN), Edge and Core networks, based on an efficient and robust Slice Selection service (virtual network selection), under an architecture that provides compatibility with the standards specification in progress (SILVA et al., 2022b). It is also applicable to a number of segments defined by 3GPP (3rd Generation Partnership Project), such as 1) the integration of a large amount of data from devices linked to the IoT context (e.g. smart homes, patient monitoring, Wireless Sensor Networks—WSNs) with cloud services; 2) the integration of applications that have multimedia requirements, high density of video and audio traffic, such as applications that use Virtual Reality (VR) and Augmented Reality (AR) technologies; 3) the provision of specialized networks with sophisticated slice selection and security mechanisms to provide services for autonomous vehicle networks in different models and topologies (e.g., V2V—Vehicle-to-Vehicle, V2I—Vehicle-to-Infrastructure, V2X—Vehicle-to-Everything) (CAMPOLO et al., 2018; RICART-SANCHEZ et al., 2018; AFAQ et al., 2020).

More than the understanding of NS, this work contributes to the improvement of the network slice selection process. A framework that implements the NSSF service in vertical and horizontal models is proposed, where the handover decision is shared by both the network edge and the User Equipment (UE) at network runtime.

1.2 Research questions and objective

The selection process requires mobile devices to discover the slices available in a given RAN, inserted in a environment of heterogeneous technologies, and make an evaluation based on a set of criteria and metrics (e.g., QoS and QoE), so as to infer which networks fulfill the continuity requirements or service improvement. In Horizontal Slicing model, the problem is multivariable, dynamic and nonlinear, in addition to having a stochastic behavior, given the randomness of variable values that need to be considered (KHEDDAR et al., 2022), (SILVA et al., 2022c), (JOŠILO; DÁN, 2020), (GLIGOROSKI; KRALEVSKA, 2019). Therefore, getting a viable model that can represent the real behavior of the slices QoS variables, as well as the RANs, is a quite complex task (KLIKS et al., 2018), (CHOI; PARK, 2017).

In addition, for Vertical Slicing Model, a given network operator may have several specialized and commonly used slices that were made available to its users. Thus, the traffic generated by these users must be aggregated, selected and forwarded to the slice that is best suited to handle and process it in the core of the network, in order to optimize their operating resources (OPEX) and comply with the agreed SLA (Service Layer Agreement) (IANNELLI; HILL; WANG, 2020), (KIM; KIM, 2019), (KIM; KIM; LIM, 2019).

Unlike the Vertical Slicing model, the Horizontal Slicing model provides a way to offer mechanisms that can take into account the Quality of Experience (QoE) and application profiles for each UE or group of UEs. This model ensures that the requirements are met in an end-to-end (E2E) approach. Additionally, it allows devices to share their resources, including communication, processing, and storage. As a result, a UE can use different slices simultaneously, both in the vertical and horizontal model (SILVA et al., 2022b), (KHEDDAR et al., 2022), (SILVA et al., 2022c).

Because of the difficulty addressing the problem described in the previous section, in addition to the need for the exploration of the different issues involving the slice selection problem in this work, several research questions have been made.

The main research questions (RQ) of this work were:

- **RQ1. "How to select the best slice?"**. The standardization entities recommend that the NSSF function consider a set of criteria or network parameters, and verify at any given time and among the available slices, which better fits the user needs, supporting network exchange (handover process) for the mobile. In this case, the process of selection slice is subject to various criteria and attributes. There is no guidance on how this should be done, and therefore, several solutions can be proposed (SILVA et al., 2022b),(SILVA et al., 2022a),(YOU et al., 2019; BARAKABITZE et al., 2020; TEAGUE; ABDEL-RAHMAN; MACKENZIE, 2019),(KARATAS; KORPEOGLU, 2019).
- **RQ2."Do the selected slices provide the necessary requirements for the user?"**. This is an important question as more mobile devices and a service variety become available in the vertical slicing model. Developing an appropriate logical architecture that optimizes the network resources provision for each consumption profile is essential. This should consider factors such as SLA, mobility, connection speed, latency, and device density while providing isolation levels that ensure independence and reduce problems with losses and errors. This avoids the problems inherent in sharing resources in traditional networks, paying for the availability and

criticality of services accordingly (SILVA et al., 2022b),(BU et al., 2019),(HABIBI; HAN; SCHOTTEN, 2017).

- **RQ3."Are the selection criteria modeled generically, regardless of the access network technology?"**. This point is key in this work. The data acquisition models for evaluating the slices must be obtained transparently, without additional computational cost for the devices. Thus, it is necessary to map the user's consumption profiles, the conditions, and the slices load in runtime for the various domains. These features allow integration and collaboration with network orchestrators from different service providers (SILVA et al., 2022b),(SILVA et al., 2022a).

- **RQ4."Does the approach employ techniques aimed at the integration and interoperability between RAN, Edge, and Core networks?"**. Unlike conventional network equipment solutions, the so-called appliances or black boxes implement all layers and are directly integrated into the hardware. In cloud-native networks (e.g. 5G and 6G), it requires the softwarization model, for example, involving Software-Defined Networking (SDN) and Network Functions Virtualization (NFV) technologies. The abstraction is possible due to several entities, such as SDN Controllers, orchestration tools, and virtualization solutions. This integration poses a great technical challenge (SILVA et al., 2022b),(BARAKABITZE et al., 2020).

- **RQ5. "Do the solutions provide compatibility with the main specifications standards under development?"**. This question shows the importance of the slice selection problem, and how the proposition of NSSF solution can allow the creation of standards in 5G industry. In this sense, several institutions and working groups (e.g. 3GPP, IETF, NGMN, ONF, ETSI, and ITU) have proposed approaches, architectural standards, and business models, constituting research and market open field. The integration and interoperability between the different solutions will only be feasible to be achieved if based on regulatory and standardization mechanisms (SILVA et al., 2022b; BARAKABITZE et al., 2020; ORDONEZ-LUCENA et al., 2018; SAADON; HADDAD; SIMONI, 2019; HUSAIN et al., 2018).

Considering these issues, new approaches to the NSSF in the vertical and horizontal model, as well as modifications of already conceived architectures and frameworks, have been formulated (SILVA et al., 2022b),(SILVA et al., 2022a),(DIMOLITSAS, 2020), (BAKMAZ; BOJKOVIC; BAKMAZ, 2020), (RIVERA et al., 2019), (BOJKOVIC;

BAKMAZ; BAKMAZ, 2019), (VINCENZI; LOPEZ-AGUILERA; GARCIA-VILLEGAS, 2019), (CHOI; PARK, 2017).

In addition, three secondary questions were made to evaluate the different aspects of the approach adopted in this work. Their results are useful not only for the telecommunications industry and major market players, but also for professionals and researchers in addressing solutions and methodologies that consider the different problems listed. These questions were:

- **SRQ1: "Is it feasible to propose a mathematical model that considers features and behavior of the different QoS variables for the NSSF function?"**. The slice selection problem is stochastic, dynamic, multi-variable, and time-varying problem. Then, the proposition of optimal models is a hard task, given three difficulties: 1) the representation of the actual behavior of the QoS and QoE variables; and 2) the obtainment of objective functions.
- **SRQ2:"Are the use of artificial intelligence techniques sufficient to predict slice behavior during the network runtime?"**. The recurrent adjustments of templates (loads, changes, and adapts), configuration parameters, and optimizers in already trained models constitute another difficulty for the proper use of the models. Besides, the abrupt changes occurring in the metrics used by the slices may or may not be associated with network anomalies, security issues, processing in nodes and/or the need for scalability cloud parameters and virtual network elements at network runtime. All these operations must be done transparently and automatically for users in the horizontal and vertical network slicing model.
- **SRQ3: "How to ensure that the selection of slices made by a given user equipment is carried end-to-end? What granularity of automation and interoperability is required to orchestrate multiple network elements across different administrative domains?"**. This is a crucial question for evaluating different solutions and constitutes a large operational issue. Note that the slice selector must be multidomain, and must address complex automation models and Machine Learning Operations (MLOps) and Software Development and Operations (DevOps) techniques, in addition work with federated SLAs. Addressing these points configure relevant and current research points.

This work presents the implementation of an approach that uses machine learning and the application of decision-making methods to assist in the NSSF service implementations and operations in multi-domain slicing environments. Its approach focuses on the evaluation and dynamic mapping QoS requirements for each type of service, user profile, and specialized slice. Initially focusing on the vertical model from edge computing, it then provides a distributed solution in the horizontal model, taking into account network access, and the signaling and message exchange between the UE and the edge framework. Thus, the following original objectives of this thesis are presented.

General Objective:

- Propose a multi-criteria approach for the NSSF service based on an architectural model for 5G and future mobile networks.

Specific Objectives

The specific objectives of this thesis is to:

1. Formulate an approach that incorporates several decision-making methods in the context Network Slicing.
2. Use machine learning algorithms in a network traffic aggregation, characterization, and forwarding model using cloud-native, NFV, and SDN techniques.
3. Propose a multiplatform framework that integrates (1) and (2) in 5G Mobile Network environments and future communication systems.
4. Validate entire approach proposed in a hybrid cloud environment and implemented framework consolidation.

1.3 Reasons for the study

In a 5G E2E architecture and future mobile networks, it is necessary to propose computational techniques that include various slice selection mechanisms, being offered to users as a transparent service. In addition, it is expected the proposition ML models for packet characterization and computation at the edge network, according to the slice profile provided by the access networks or user preferences, or even contracted by telecommunications operators from their administrative domain to the public cloud (SHI et al., 2021), (JOŠILO; DÁN, 2020), (CHEN et al., 2020),(ZHANG et al., 2020).

The main focus is to propose a slice selection strategy that involves sharing the decision model between the UE and the 5G/B5G network, which is called horizontal slicing. This strategy should complement the proposals in the vertical slicing model, implemented regardless of whether common or specialized slices. It is important to note that in the vertical model, Telcos use network orchestration tools to create and maintain these slices (SILVA et al., 2022b).

It is also expected that the new approach uses efficient strategies, coupled with SDN and NFV control functions, that comply with the main specification standards under development (especially those led by 3GPP). These points constitute an open field of research and solutions for the market (ZHANG et al., 2020), (ORDONEZ-LUCENA et al., 2018), (MORGADO et al., 2018), (HABIBI; HAN; SCHOTTEN, 2017).

Thus, the Network Slicing architecture, with a well-defined and robust Slice Selection service, emerges as the main solution for the next-generation mobile networks, enabling integration and adaptation in different segments and applications. This is an extremely relevant problem, as there is still no defined and marketable industry standard (ZHANG et al., 2021), (DIMOLITSAS, 2020),(SAADON; HADDAD; SIMONI, 2019), (HUSAIN et al., 2018).

1.4 Main contributions

The slices selection task on network runtime on heterogeneous environment is a difficult problem, since there is still no fully accepted solution or technique in this field. The reason is that there are many variables and scenarios, as the solutions that consider or not the inter-slice mobility process.

Thus, the implementation of new slice selection techniques becomes quite viable, including the demand in the growing use in smart cities, robotics, agriculture 4.0, healthcare, remote surgery, Unmanned Aerial Vehicle (UAV), IoT, and Internet of Vehicles (IoV), among other technologies and scenarios involving network convergence.

Specifically, the points presented below are the present work's original contributions to its field of study.

1. Proposal of an NSSF service framework in both vertical and horizontal models, where the handover decision is shared by network edge and user's equipment on network runtime;

2. Suitability of NSSF function originally from 5G core to network edge, to UE, or to 5G Radio Base Station. The deployment model could be defined according to the

institutional operating scenario;

3. Slice selection model implementation based on multicriteria decision techniques, fuzzy logic and machine learning, using hybrid algorithms and MLOps concepts;
4. Strategy formulation independently of packet marking type (*e.g.* SR-IPv6, MPLS, VXLAN, VPN, EVPN, H-QoS, and GTP) for analyzing data flows and forwarding to available slices;
5. Approach for collecting QoS and QoE metrics directly from the TCP/IP stack, and signaling the UE profile, without legacy infrastructure modifications;
6. Definition of network datasets characterization and acquisition models, in addition to decision matrices assembly in network runtime, using data analytics tools;
7. Integrated model specification and implementation with market orchestration tools such as ONAP (Open Network Automation Platform), OSM (Open Source Mano), and EMCO (Edge Multi-Cluster Orchestrator);
8. Slice selection embedded application development for multistream mobile devices.

1.5 Document structure and chapter overview

This work is organized other six chapters.

Chapter 2 presents an 5G overview and future networks, highlighting the slice selection problem in the context of network slicing. Besides, it demonstrates how edge computing solutions can be adopted to solve the problem. Finally, it presents the main techniques, methods, and tools that make up the solution conceived in this work.

Chapter 3 concentrates on a review of the literature regarding the research problem. A table comparing the main related works is presented, discussing the relevant points of the adopted solutions, as well as their limitations.

Chapter 4 presents the NSSF DAF framework architecture, its modules description, the framework technical specification, and an integration overview of 5G networks and future communication systems. In addition, it presents the solution limitations and restrictions, as well as its scalability and extensibility capacity.

Chapter 5 contains the experiments scenarios description for proposed solution validation, as well as the description of the tools and technologies used in the tests; there are also details of the main proposed algorithms.

Chapter 6 reports and discusses the results obtained. A detailed statistical analysis is carried out for the experiments conducted in different scenarios. A comparison between the methods is also presented, seeking to evaluate which method presented greater accuracy in the selection of slices.

Chapter 7 concludes the work. In addition, the answers to the research questions listed in this work are resumed and analyzed. The main contributions of this work, research limitations, and recommendations for future work are presented.

2 FUNDAMENTAL CONCEPTS

2.1 5G Vision and Network Slicing

Telecom operators and standardization bodies are working to fully implement 5G networks by the year 2024 (3GPP Release 18)[1]. Among the main innovations expected by these networks is the possibility of delivering network slices based on the user's consumption profile. This way, the network should be prepared to meet the demands to meet specific applications and scenarios demands, such as smart cities, vehicular networks, patient monitoring, and medical care. Moreover, these networks should enable the integration of the aforementioned applications to cloud services (MORGADO et al., 2018), (KARATAS; KORPEOGLU, 2019).

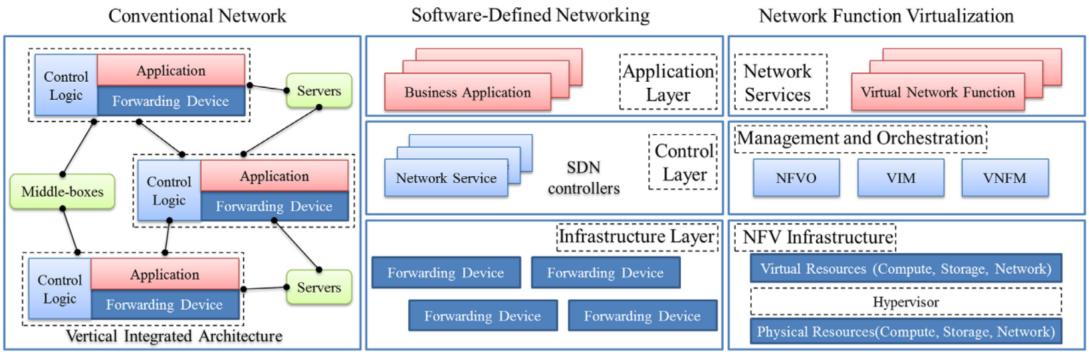
Thus, it is expected, for each consumption profile, the delivery of optimized network resources considering the SLA and satisfying requirements such as connectivity, connection speed, latency, and device quantity, among others. In addition, to offering isolation levels that allow independence, i.e. computing and network resources competition-free, reducing problems with losses and errors, issues that are inherent to the infrastructure sharing in traditional networks (BONATI et al., 2020), (BARAKABITZE et al., 2020).

Virtual networks architectures, associated with traditional networks and core networks from service providers, make up the core concept of Network Slicing. A Network Slicing architecture uses resources from SDN and NFV. These two technologies enable the partitioning of traditional network architectures into multiple virtual networks on top of a single physical network infrastructure, which is shared by different users and application profiles(KIM; KIM, 2019),(HABIBI; HAN; SCHOTTEN, 2017).

Figure 1 presents an architectural overview of these technologies compared to traditional ones. It is worth highlighting that, when it comes to solutions involving SDN and NFV, there is an evident division between layers that make up the softwarization model. Unlike conventional network equipment solutions, the so-called appliances or black boxes

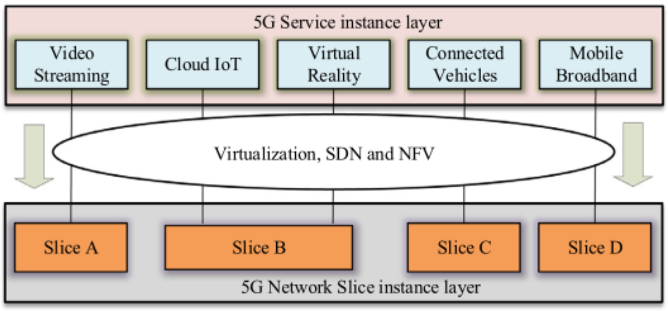
[1]https://www.3gpp.org/specifications/67-releases

Figure 1 – Comparison between conventional, SDN and NVF networks.



Source: (BARAKABITZE et al., 2020).

Figure 2 – Networking Slicing Concept.



Source: (YI et al., 2018).

implement all layers and are directly integrated into the hardware. Each layer implements a level of abstraction from the physical infrastructure layer to the application layer. This abstraction is possible due to several entities, such as SDN Controllers, orchestration tools, and virtualization and hypervisor (BARAKABITZE et al., 2020).

In this context, a particular consumer may require a service that offers very high speed, low delay, and no data loss, paying accordingly for the availability and criticality of its services. The approach adopted in this work allows the setup of dedicated and flexible networks to meet a specific type of user. Figure 2 presents the concept of network slicing from the perspectives of SDN and NFV technologies. Note that each application (e.g. Video Streaming, Cloud IoT, Virtual Reality, Connected Vehicles, and Mobile Broadband) uses a specific slice (YI et al., 2018).

To enable network resource abstraction on the same physical hardware, it is necessary to separate the data plane, control, and transport in the network equipment (e.g. routers, switches, access points). These terminologies are characteristic in SDN networks, and configure the real possibility of handling traffic data in certain equipment under different perspectives. Fundamentally, management functionalities, as well as data behavior programming, based on rules, service delivery, and orchestration model, make it possible to expand network capabilities. In addition, they allow innovative and creative applications to emerge for the so-called internet of the future (SAADON; HADDAD; SIMONI, 2019), (ORDONEZ-LUCENA et al., 2018).

Several manufacturers have supported and collaborated with different standardization groups and research labs in order to define the technical requirements for the full implementation of 5G mobile network resources in combination with the Network Slicing architecture. This combination aims to substantially guarantee SLAs from operators, in addition to offering customizable networks, meeting the QoS requirements per application, and ensuring the best possible QoE for users (HUSAIN et al., 2018), (BONATI et al., 2020).

Thus, the Network Slicing concept is pointed out as the path to new mobile network solutions, including 5G and future networks. However, although several proposals indicate promising paths, it is not yet possible to add the various features in a fully functional approach. A satisfactory solution must define the operation and management mechanisms for each slice, in addition to providing scalability, orchestration, and support in a multidomain decision environment, a task that involves heterogeneous network technologies (MORGADO et al., 2018).

Finally, we believe that the proposal of technical alterations in mechanisms, models, and NS architecture functions defined by 3GPP is innovative. These modifications attempt to support 5G and future communication networks. In this context, improvements in Slice Selection mechanisms will be relevant for both research and market solutions. In order to accomplish this goal, approaches that take advantage of protocols and computational tools for classification, composition, and selection of slices through machine learning, and decision-making methods in a completely virtualized architecture with SDN and NFV will be key components in this field (SILVA et al., 2022b),(SAADON; HADDAD; SIMONI, 2019),(ORDONEZ-LUCENA et al., 2018),(HABIBI; HAN; SCHOTTEN, 2017).

2.2 Intelligent Edge Computing

Edge computing refers to a broad set of techniques designed to move computing and storage out of the remote central cloud (public or private) and put them closer to the source of data (YAN; SHENG, 2023),(ETSI, 2019). It aims to increase the computational capacity of the network since next-generation networks must support the connection density of more than 1 Million devices/km^2.

New applications, such as tactile Internet services, may also require an extremely high data rate, lower latency, and extended reliability. To facilitate these types of services, edge computing is incorporated as a technique to process data at the network edge, instead of transferring it to the cloud network that is far away.

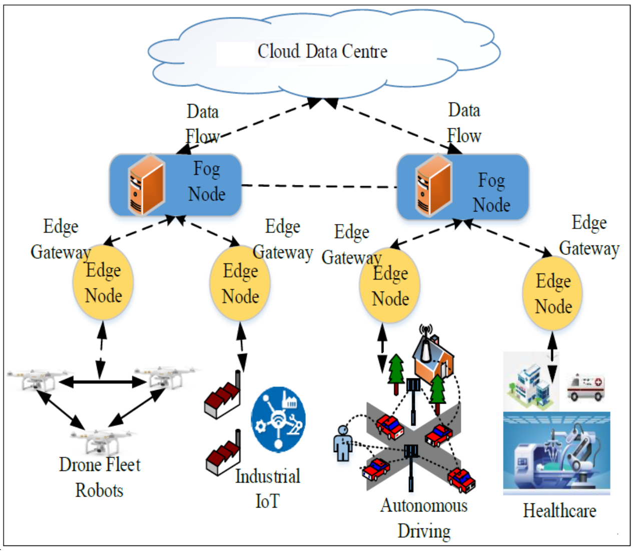
Within the broad topic of edge computing, ETSI Multi-Access Edge Computing (MEC) is the widely accepted standard that must be met for a technology to be considered edge computing (MOURADIAN et al., 2018). Its standards are guided by the following principles (RAJ et al., 2021),(R; MOHANA, 2022),(ETSI, 2019):

- Edge computing must have a virtualization platform to be considered MEC (ETSI uses its NFV architecture in the standard);
- MEC can be deployed at radio nodes, aggregation points, or collocated with the Core network functions;
- APIs (Application Programming Interface) in a MEC environment must be simple, controllable, and, if possible, reusable for other tasks;
- Since the computing, storage, and network resources required by an MEC application may not match what is available in a node, an MEC network requires lifecycle management for the entire application system to handle these variables correctly;
- MEC systems must be able to relocate a high-end mobile application running on an external cloud to an MEC host and vice versa, meeting all the application requirements (ETSI admits that this principle needs further research).

Figure 3 shows the cutting-edge computing technology used to enable several next-generation applications. Differently from fog computing (a concept formulated by Cisco (CISCO, 2022)), edge computing is associated with last-mile processing. A fundamental difference between MEC and fog computing is that MEC works only in the autonomous mode, whereas fog computing has several interconnected layers and can interact with the distant cloud and the network edge (DOGRA; JHA; JAIN, 2020; CHIANG; ZHANG, 2016).

The combination of Artificial Intelligence (AI) and edge computing was introduced to manage several emerging future communication problems. However, with the limited

Figure 3 – Edge computing and Fog computing interaction with some applications.



Source: Based on (DOGRA; JHA; JAIN, 2020).

availability of resources and storage, it is very difficult to operate highly complex AI-based algorithms that require huge data collection at endpoints. Research is needed to design new and thin AI algorithms for edge nodes. Moreover, the development of effective mobile resource scheduling and transferring techniques to improve system performance is also necessary (RAJ et al., 2021),(YOU et al., 2021).

Thus, by combining edge computing with AI and NS, we are able to meet the latency requirements of critical services, ensuring efficient network operation and service delivery with improved user experience. Furthermore, intelligent edge computing is important for slice selection as it is able to alleviate the network core functions. In this sense, Section 2.3 presents the slice classification techniques, and Section 2.3.2 discusses slice composition with ML.

2.3 Slices classification techniques

Network selection in a heterogeneous environment represents a difficult problem, since there is no solution or technique accepted in this field yet, due to the number of variables and existing scenarios. Evidence of this can be found in the solutions that do or do not consider the process of inter-network mobility.

The implementation of new techniques for network selection thus becomes quite feasible, mainly because of the growing demand for its use in vehicular networks, patient monitoring, and smart cities, among other technologies and scenarios involving network convergence (CAMPOLO et al., 2018; RICART-SANCHEZ et al., 2018; SALAMA; SAATCHI, 2019).

The most common methods reported in the literature for solving the problem of network selection include the usage of fuzzy logic, Multiple Attribute Decision Making (MADM), Multiple-criteria Decision Analysis (MCDA) or Multiple-criteria Decision-making (MCDM), Genetic Algorithms, Artificial Neural Networks, and ML. Among the employed MADM methods are the following: the Analytic Hierarchy Process (AHP), Simple Additive Weighting (SAW), TOPSIS (FIGUEIRA; GRECO; EHRGOTT, 2005; RIOS; MONTEIRO; GONDIM, 2012), Multiplicative Exponential Weight (MEW), Simple Multiattribute Rating Technique (SMART) (TEAGUE; ABDEL-RAHMAN; MACKENZIE, 2019; KARATAS; KORPEOGLU, 2019), VIKOR, COPRAS, Promethee II, and Grey Relation Analysis (GRA) (BOJKOVIC; BAKMAZ; BAKMAZ, 2019; SALAMA; SAATCHI, 2019; SHI et al., 2021; WANG et al., 2020).

With regards to models that consider hybrid solutions, a feature that has achieved good results is concerned with the techniques that include fuzzy logic, MCDM methods, and ML (CHAHAL; HARIT, 2019; AFAQ et al., 2020; TEAGUE; ABDEL-RAHMAN; MACKENZIE, 2019; ZHANG et al., 2020; GLIGOROSKI; KRALEVSKA, 2019; RIVERA et al., 2019; SHURMAN; RAWASHDEH; JARADAT, 2020; LU et al., 2020; SANTOS et al., 2020). These models generally work in a similar way. After the process of data collection, according to the criteria outlined in the previous section, fuzzy logic processing occurs, followed by the classification method via a decision strategy. In this case, for each criterion, a certain weight is added to prioritize some services over others, guiding the choice of network in accordance with the application in use.

2.3.1 Multicriteria methods

Multicriteria decision-making approaches consist of several techniques used for solving problems. However, there are uncertainties, information conflicts, and disputes over the criteria that are required to evaluate the alternatives. These characteristics are inherent parts of the problem, and must therefore be mapped by the decision maker (person or system) that will solve it (WĄTRÓBSKI et al., 2019). The aim is to assess several possible options using various criteria, each with its own set of characteristics. These characteristics may include the weights, degrees of importance, or preferences of the decision maker (HEZER; GELMEZ; ÖZCEYLAN, 2021).

- The Promethee (Preference Ranking Organization Method for Enrichment Evaluations) II method applies a concept called overclassification, which is a multi-attribute decision-making method proposed by the European School. Thus, the alternatives are compared in pairs in order to infer whether a given alternative is as effective as the other. If alternative a is better than alternative b according to a given criterion, a is said to overclassify b. When all alternatives are compared, the final result is a classification (ranking) of all the alternatives in a given set, from best to worst. For this work, the second version was chosen, as Promethee I includes situations where alternatives are incomparable. Promethee II does not make judgments if the alternatives are incomparable, so that it always allows for a complete classification of actions (ALINEZHAD; KHALILI, 2019; FIGUEIRA; GRECO; EHRGOTT, 2005).

- The TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) method employs the principle of choosing an alternative that is closest to the positive ideal solution (best solution), and farthest from the negative ideal solution (worst solution). Thus, the method focuses on the maximization of the benefits and the minimization of the costs (BAKMAZ; BOJKOVIC; BAKMAZ, 2020). The method has been widely used and described in the literature for different problems and has several extensions, in addition to hybrid models with other techniques (BOJKOVIC; BAKMAZ; BAKMAZ, 2019).

- The VIKOR (Visekriterijumska Optimizacija i Kompromisno Resenje) method is based on the concept of compromise ranking. It means that it defines a measure of proximity with the ideal solution. In this context, the method uses a linear combination of Manhattan distance and Tchebychev distance metrics, where the former represents the maximum group utility, and the latter represents the minimum individual weight of the "opponent". After obtaining the decision matrix and the weight vector, the method employs a set of mathematical operations, resulting in the ranking of the alternatives (ALINEZHAD; KHALILI, 2019; WĄTRÓBSKI et al., 2019; BABASHAMSI et al., 2016).

- The COPRAS (Complex Proportional Assessment) method seeks to assess the alternative superiority by implementing a ranking that considers their performance based on different criteria and their weights. The method can be used to maximize or minimize criteria, and it is used in several areas of knowledge (HEZER; GELMEZ; ÖZCEYLAN, 2021; ALINEZHAD; KHALILI, 2019).

- The MABAC (Multiattributive Border Approximation Area Comparison) method was formally proposed in 2015, and its main feature is the ability to assess the distance between each alternative and the Border Approximation Area (BAA). The method has simple operations and calculations; however, the results are stable and it considers the relationships between gains and losses in its judgments, which allows for a 'reasonable' decision-making process. These characteristics allow its use in conjunction with other methods, resulting in more refined solutions (WANG et al., 2020).

- The CODAS (Combinative Distance-based Assessment) method uses two measures to assess desirability or preference among alternatives. The first measure consists of calculating the Euclidean distance of the ideal-negative solution, which includes an indifference space between the criteria. In this way, the alternative whose distance is greater than the ideal-negative solution is preferable. In cases where it is not possible to use the Euclidean distance due to the alternatives being incomparable, the Taxicab distance is used as a secondary measure (OUHIBI; FRIKHA, 2020).

- The OCRA (Operational Competitiveness Rating) method was formalized in 1994, and has been applied in several areas ever since. The method consists of a relative performance measurement technique, and is based on a non-parametric model. The main feature is the ability to monitor and compare the performance of decision-making units (DMUs) over time. In order to accomplish this goal, the method considers 06 steps, and at the end of the process, it ranks the alternatives in descending order through the use of efficiency indices (ULUTAş et al., 2020), (KUNDAKCı, 2019).

- The CoCoSo (Combined Compromise Solution) method uses the evaluation of compromise between the alternatives, aiming to evaluate the benefit and cost criteria for the problem under analysis. Moreover, the method uses a comparability verification model between the alternatives that consider the weights, the power of the weights, and the aggregation strategy. At the end of the operations, the alternatives are ranked according to their significance (YAZDANI et al., 2019).

- The ARAS (Additive Ratio Assessment) method implements the concepts of utility theory. In this case, the decision matrix undergoes a set of normalization operations, after which the criteria weights are applied, and finally, the optimality function is calculated. The optimality function determines the utility degree of each alternative, which considers an interval between 0% to 100%. At the end of the process, the alternative that presents the greatest utility is chosen as the best among its counterparts (ALINEZHAD; KHALILI, 2019).

- The MARCOS (Measurement of Alternatives and Ranking according to Compromise Solution) method is also based on utility theory. The method application involves 07 steps. Initially, two columns containing the ideal (AI) and anti-ideal (AAI) solution are added to a decision matrix. Afterward, certain normalization operations and a maximization and minimization process are performed according to a criterion. Regardless of the criterion, the method has proven to be effective. The decision matrix is multiplied by the weight vector of the criteria, resulting in a weighted matrix. Finally, the calculation of the utility function is performed, aiming to determine the degree of utility of each alternative in the system. The alternatives are first ranked based on the solutions provided by utility function, and then sorted considering their final values (NGUYEN et al., 2022a), (NGUYEN et al., 2022b).

- The MAIRCA (Multi-Attributive Ideal–Real Comparative Analysis) method uses the same initial operations as the MARCOS method in the decision matrix. However, it is necessary to calculate the preference for each alternative based on their weights, thereby obtaining a "theoretical rating matrix", which is based on the "real rating matrix". A set of operations is then performed to determine the criterion functions (Qi) final values. The final ranking is obtained using the sum of gaps (g_{ij}) (NGUYEN et al., 2022a), (NGUYEN et al., 2022b).

- The EDAS (Evaluation based on Distance from Average Solution) method is widely used in problems with stochastic characteristics. In general, the method is based on the distance calculation between the alternatives and the optimal value for obtaining the best alternative. The attribute average solution is first calculated based on the decision matrix, and then the positive and negative distances average solutions are obtained. At the end of the process, the attribute weights are applied over the distances. The normalized matrix is determined in order to evaluate each alternative, ranked in descending order (ALINEZHAD; KHALILI, 2019).

- The SPOTIS (Stable Preference Ordering Towards Ideal Solution) method uses the normalized distance calculation of each alternative in the decision matrix, as a function of the ideal or best solution. The operations set uses several formulas, aiming to reduce the average distance to an ideal multi-criteria solution, that is arranged in ascending order. It is important to emphasize that the method evaluates each alternative independently of the others, which means that it is not possible to invert the classification (DEZERT et al., 2020).

- The MOORA (Multi-Objective Optimization on the basis of Ratio Analysis) method uses a pipeline similar to the other MCDM methods, which consists of i) defining decision matrix; ii) normalization process of the decision matrix; and iii) attributes optimization from a set of equations. However, in several works, the MOORA method presented results with better performance, and low computational cost when compared to the others (NABABAN et al., 2021; SUNGKONO et al., 2022). The justification for this is the fact that the method was designed to remedy the weaknesses of previous methods. Thus, the method works simultaneously with several problem restrictions, while optimizing the different attributes and criteria, presenting a good level of precision in the alternatives ranking, and facilitating the decision-making process (SAHIDA; SURARSO; GERNOWO, 2019).

The methods selection for the framework validation that this work proposes considered their popularity in the literature for different types of problems, applications, and areas of knowledge, as well as their proven efficiency. The details of some of these methods and their mathematical operations are presented in Appendix A. They are used as a reference in the application of the other methods used in this work.

2.3.2 Slice composition with machine learning

The resources allocation for slice composing must support its lifecycle management function (TALEB et al., 2019), which, in turn, must meet the requirements specified in the SLA and in the QoS and QoE rules as latency metrics, throughput, and capacity of available resources (VASILAKOS et al., 2020). Considering distributed computing (JR et al., 2019), virtual systems and functions, the variable dynamics of the network, as well as the dynamicity and variance of the parameters over time, make the resources orchestration for slice composition a complex task for applying a solution based on AI (JR. et al., 2021). However, a premise to apply AI techniques is to have an accurate set of data. Therefore, it was necessary to build or search for datasets that portray the slice efficiency metrics or even use data streaming for application to AI models.

Hence, monitoring a network's incoming traffic or its historical behavior in the form of raw data by streaming can represent the network behavior in terms of QoS and serve as a basis for obtaining optimal models. The research interest in resource allocation usually focuses on how to implement slice instances according to the description of resource requirements (GUAN; ZHANG; LEUNG, 2020) and following the service agreement.

In this sense, a methodology for data collection and analysis from flexible network that supports the interactive and iterative model for producing ML Models, is validated.

Thus, a good strategy is to establish a data format to train various AI models and perform slice recognition and recommendations given the flow generated by the client.

The research interest in resource allocation focuses on how to implement virtual network instances under shared physical resources. End-to-end NS for 5G mobile networks using ML has been intensively studied (TALEB et al., 2019; VASILAKOS et al., 2020; BARAKABITZE et al., 2020; LI; OTA; DONG, 2019; LIU; HAN; MOGES, 2020; SHI; SAGDUYU; ERPEK, 2020; USAMA et al., 2020). One of the most important issues related to the use of slice by attribution performed from ML is to guarantee the generation of optimal models to avoid the resources sub-utilization as well as the SLA breaking.

Establishing metrics and an optimal slice recovery mechanism for deterministic demands was the starting point in the work of Wen (WEN et al., 2019), in which he sought to build the ideal solution to be used as a reference in other algorithms. In stochastic demands, the optimization mechanisms option that generally has slow convergence is initially adopted, with the use of ML as a future solution. Establishing a dataset that reflects an optimal solution for guiding ML is an appropriate scenario for building supervised learning models. However, scenarios diversity and demands networks clients make standardized definition a complex task.

Thus, exploring learning types that are not dataset dependent containing the optimal model classifier label is presented as a plausible and viable alternative. Hence, for instance, unsupervised learning, recurrent, and convolutional neural network models are often used to build NS solutions. In the work by Toscano (TOSCANO et al., 2019), neural networks of the Long-Short Term Memory (LSTM) type were used for slice provisioning in a simple mechanism with only four parameters contributing to the SLA, being Rs, Ks, Ws, and Ds, where Rs is the average ratio of resource utilization, where $Rs \epsilon [0, 1]$, Ks measures the maximum standard deviation of resource sharing, Ws is a time window in which the average is calculated and where the resources sharing plus the deviation must be guaranteed, and Ds specifies the lifetime duration of the slice. The neural network training was conducted by using data collected from 24h of running a traffic simulation in the NS-3 Network Simulator (NS-3, 2022). After deploying the performance achieved during the training of the network, the author concluded by saying that more data needed to be generated to supply the LSTM network.

Cui (CUI et al., 2020) explore a variation of the LSTM architecture together with Convolutional Neural Networks (CNN) in a slice context approach for vehicular networks (V2X). The neural network training used a mobile network traffic dataset from the city of

Milan, Italy (BARLACCHI et al., 2015), which contained data from the following three categories: SMS (Short Messaging Service), telephone, and web browsing. For the simulation, each category considered a slice to be allocated through the model which, in turn, presented a satisfactory performance in establishing the connection between the customer and the slice that would serve them. The LSTM network is of the Encoder/Decoder type, and the encoder was used to predict the network traffic, while the decoder obtained the optimal slice. The entire dataset was used for network training, with the input in the network being an image matrix. Despite the satisfactory result reported, neither the SLA nor the parameterization of the dataset attributes used was presented. Therefore, replicating the procedure to validate the network performance becomes impractical.

2.3.3 Fuzzy logic

Discussed in detail in Chapter 4, the solution proposed in this work initially concentrated efforts on mapping the QoS requirements, as well as the SLA, as the central problem of the slice selection process. In this context, the proposed solution approach also made use of fuzzy logic to assist in the QoS and QoE measurement and evaluation in 5G mobile networks, since it can deal with uncertainties and subjectivity, having its use widely disseminated in these scenarios (MORDESON; MATHEW, 2018).

Fuzzy logic consists of different approach from classical (Boolean) logic, that is, it allows for the treatment of variant and subjective values, such as the ones between 0 and 1. Thus, it is able to ponder the pertinence of values 0.1, 0.5, and 0.9, being almost false, half true, and almost true, respectively. This treatment emerged as a way to deal with complex situations, and mainly to deal with uncertainties (CASTILLO; CASTRO; MELIN, 2022). The fuzzy logic application to slice selection problem is detailed in section 5.2.2.

The resources and concepts used can be divided into four parts:

1. Fuzzy Input: a set of QoS data collected in the following Test Setup 5.2.2 was considered. The variables are: Latency, Jitter, Loss, Bandwidth, Transfer, Distance and Reliability.

2. Fuzzification: the fuzzification process used triangular, trapezoidal and Gaussian membership functions. The input values are normalized, and the parameters organized into three linguistic variables, namely: low, medium and high. The fuzzy operators are applied using the defined linguistic variables.

3. Inference Engine: the inference engine consists of a set of rules that seek to find the relationships between the input and the output values of the system; therefore, they are applied directly to the data set and use the Mamdani inference mechanism. These rules constitute the knowledge base for the problem under study.

4. Desfuzzification: this process returns evaluated values by the inference model, transforming them into numerical and intelligible information again. This process is related to the system output and essentially composes a classifier.

A detailed fuzzy logic use explanation in different scenarios is outside the scope of this work. However, several examples of its application and case studies can be found in (CASTILLO; CASTRO; MELIN, 2022),(MORDESON; MATHEW, 2018),(MATHEW; MORDESON; MALIK, 2017).

2.4 Chapter summary

This chapter described the main computational techniques addressed to the slice selection problem. A brief explanation is given about each one, and how they correlate with the problem under investigation. In addition, fundamental concepts and technologies that make up the 5G framework and future networks are presented.

The details of these concepts are crucial to understand the solution for which this work argues and to obtain answers to the research questions raised in the previous chapter. In this sense, the adoption of the mentioned technologies and techniques, with the developed algorithms, constitutes a feasible way to solve the problem and contributions claimed in this work.

3 LITERATURE REVIEW

3.1 Related Work

The purpose of this section is to contextualize the slice selection problem in the scenario of heterogeneous mobile networks (e.g. 5G and Wi-Fi 6) to support the contribution of this work, which is given in Chapter 4.

Several solutions have been proposed to evaluate and select the best slice, considering the QoS requirements for telephony and data services. In general, the literature suggests approaches that consider the following situation: given a set of criteria or network parameters, the works verify, at any given time and among the available slices, which better meets the user needs, supporting network exchange (handover process) for the mobile (YOU et al., 2019; BARAKABITZE et al., 2020; TEAGUE; ABDEL-RAHMAN; MACKENZIE, 2019),(KARATAS; KORPEOGLU, 2019). In this case, the process of choosing the slice is subjected to a number of criteria.

A higher number of mobile devices, as well as the variety of services in the vertical slicing model, require the optimized development of an appropriate logical architecture, which allows for scalability, energy efficiency, and simplification of network functions; they also provide a business model (CAPEX and OPEX) that uses the computational infrastructure in operation. Thus, the Network Slicing architecture, with a well-defined Slice Selection service, emerges as the main solution for the next-generation mobile networks (BARAKABITZE et al., 2020; HABIBI; HAN; SCHOTTEN, 2017; BU et al., 2019).

Most papers focused on strategies that offer telecommunications operators mechanisms that can provide and control resources for each virtual instance. These include customization to meet the specific user's requirements in a pre-defined structure of the resource allocation. In other words, it occurs without any choice of the slice being passed by the end-user, ignoring the type of RAN to which it is linked.

These points raise interesting discussions that need to be posed, such as: 1) the E2E model in NS environment; 2) service continuity guarantee in a roaming NS architecture, without the interruption of active services, and stability in the data delivery warranty, in scenarios with medium and high mobility; 3) interoperability between various network slice architectures under different administrative domains; 4) security in the transmission sensitive data; and 5) the migration of all physical network functions to logical networks (NFV) abstractions (MORGADO et al., 2018; KIM; KIM, 2019; AFAQ et al., 2020; CONDOLUCI; MAHMOODI, 2018).

According to (MORGADO et al., 2018; ORDONEZ-LUCENA et al., 2018; SAADON; HADDAD; SIMONI, 2019; HUSAIN et al., 2018), the points listed above are important and have led several research laboratories and institutions to seek for the strategies and solutions standarlization. Those institutions and working groups include, but are not limited to, the Internet Engineering Task Force (IETF) (IETF, 2023), Next Generation Mobile Networks (NGMN) (NGMN, 2023), Open Networking Foundation (ONF) (ONF, 2023), 3GPP (3GPP, 2023), and the European Telecommunications Standards Institute (ETSI) (ETSI, 2023). Therefore, new proposals of approaches, as well as modifications to existing architectures, constitute an open field from the point of view of researchers and the market (BARAKABITZE et al., 2020).

An issue that has arisen in some countries is whether the 5G network slicing will be consistent with the network neutrality regulations. Some say that the practical implications for current open Internet rules are speculative at this stage concerning 5G. That is because the different 5G elements, such as NS, depend not only on the occasional technological capabilities, but also on the market demand, the degree of competition, the commercial strategies, and other variables (OECD, 2019).

There are several literature contributions for admission control, resource allocation, and billing mechanisms in virtualized wireless networks (SILVA et al., 2022a), (JR et al., 2019). However, the automated mechanisms for slicing and monetization in 5G are a theme open to discussion (VINCENZI; LOPEZ-AGUILERA; GARCIA-VILLEGAS, 2019).

Relevant scientific production performed on NSSF is summarized and discussed in the following Table 1. It is worth noting that papers present contributions in different segments, which goes from data plane slice selections strategies to the use of certain decision-making methods, genetic algorithms or Bayesian networks. The proposals lack broad validation and testbed implementations in different scenarios, which comply with the specifications of standardization bodies and meet the main requirements eMBB, URLLC and mMTC scenarios.

Moreover, none of the solutions mentioned offer a deployment model that enables integration in running networks; or provide mechanisms for automation and integration with cloud and network orchestrators; or even allow models re-training based on the use of hybrid algorithms, which would provide greater extensibility and scalability in their solutions.

In this way, the framework proposed in this work, implements the NSSF function in order to cover all these requirements, filling the gaps left by previous works, and constituting a promising solution for 5G and future networks.

In addition to the Virtual Network Function (VNF) for selecting slices, Rivera, Khan, Mehmood and Song (RIVERA et al., 2019) argue that management system is also required to provide, update, and control the slices life cycle by VNFs. They focused on the provision and deployment of Data Plane Network Slices and the selection of a proper slice by taking identification parameters from the user's equipment. They also confirmed that traffic control rules can be deployed with minimal resources, so that the efficiency Data Plane slices are not affected. As the proposed system evolves, a new degree of automation can be developed with a monitoring agent that can supervise the status of User Plane of the Packet Data Network Gateway (SPGW-U) traffic in real-time.

The works in (BOJKOVIC; BAKMAZ; BAKMAZ, 2019; BAKMAZ; BOJKOVIC; BAKMAZ, 2020) proposed NSSF based on Technique for Order Preference by TOPSIS. According to the division network principles, a mobile terminal may choose from several connectivity alternatives available based on criteria related to slicing performance, service requirements, and user preferences. NSSF is perceived as a logical evolution from the ABC (Always Best Connected) concept to 5G and beyond mobile systems. NSSF is modeled as a MCDM problem exploring the principle of TOPSIS for classifying the available slices based on their attributes and weights. TOPSIS is a widely accepted decision-making tool, considering its understandable logic, algorithmic logic, and mathematical form. However, it fails to provide consistent results due to the phenomenon of rank reversibility.

As for the TOPSIS method, the results in (BOJKOVIC; BAKMAZ; BAKMAZ, 2019) demonstrated that the standard deviation weighting technique (SD-TOPSIS) presents significantly better performances regarding rank reversibility, but it is highly complexity. In this sense, they suggested applying the entropy weighting technique (E-TOPSIS), especially when faced with characteristics such as the fine granularity of slices. The authors also concluded that the improvements in TOPSIS methods are closely related to alternative methods for normalization and ranking phases, especially when managing a situation of practical implementation. As for future research, they address the need for analyzing the influence of several alternative standardization techniques and classification methods on the NSSF performance.

Besides, the results from (BAKMAZ; BOJKOVIC; BAKMAZ, 2020) take into consideration the three stages of the decision-making process (normalization, weighting, and ranking), and show that proposed alternative techniques, such as linear normalization (MAX-MIN), the weighting of variance, and binary classification alternatives, presented a positive influence by significantly reducing classification reversibility and computational complexity among the tested scenarios. This justifies the need for considering MCDM methods as a potential solution to the network slice selection problem in 5G mobile systems and their future generations. As a suggestion for future studies, the performance of other MCDM algorithms should be analyzed in terms of ranking reversibility.

The work in (SHURMAN; RAWASHDEH; JARADAT, 2020) presented a new approach that aimed at increasing service utilization and the efficiency of network slices, since slice selection is one of the key elements of 5G Network Slicing. In the proposed method, each network slice is considered as a different service and is represented by a website system with its own database. Then, the GET method is used to link the user equipment to multiple systems at the same time, focusing mainly on passing the required parameters by URL. By using this approach, Multiple-Service UE will be able to connect simultaneously to several networks, obtaining a session on the related networks for a specific time. As future studies, the authors suggested providing the user with a PURE connection to these networks, meaning that, by passing the parameters to the new system, the users could be registered to all existing systems, benefiting from their full capabilities, instead of having a temporary connection as considered in the present work.

Dimolitsas (DIMOLITSAS, 2020) presented a multicriteria decision framework for the optimal selection of Edge Points of Presence (EPoPs) to deploy a network slice. The EPoP Selecting Framework is composed of three main components: (i) the Service Registry, which contains the required KPI values for each EPoP candidate; (ii) the Filtering Engine, which is responsible for the initial filtering of the candidate EPoPs according user's requirements; and (iii) the PoP Ranking Mechanism, which selects the most appropriate EPoP for slice deployment based on the multi-criteria Fuzzy Analytic Hierarchy Process (FAHP) method. The proposed framework was assessed under a realistic scenario in comparison with simple filtering and the single-object (FAHP) approach. The results indicated the relevance of the proposed two-stage method in meeting the user's requirements for hardware and software, allowing for the communication between slices and optimal resource allocation from the provider's point of view.

(ZHAO et al., 2020) worked on maximizing system resource utilization while guaranteeing the satisfaction degree among users by exploring the E2E network slicing problem. Using theoretical analysis, the authors proved that it is a NP-hard problem, and then they proposed a Genetic Algorithm (GA) to solve the optimization problem. A simulation experiment was conducted to validate the proposed GA algorithm, showing that this method obtained better access and transmission performance when compared to traditional selection methods based on the Received Signal Strength (RSS) or greedy algorithms.

Otoshi et al. (OTOSHI et al., 2021) proposed dynamic slice selection by learning to recognize the rough application situation and the mapping between the current application situation and the future slice. The Bayesian Attractor Model (BAM) was used to achieve consistent recognition, as well as the Dirichlet Process Mixture Model (DPMM) to achieve automatic attractor construction. Situations mapping was also automatically learned using feedback. The video streaming was used as application of dynamic slice selection and the results show that the proposed method can maintain a high quality of video streaming. Extended BAM can be used to reduce the number of slice changes while reducing degradation of video streaming quality. Additionally, the integration and deletion of attractors was used to maintain only the necessary number of attractors.

Silva et al. (SILVA et al., 2022a) presented a solution for 5G network slice selection in IoT scenarios. It uses Edge Computing resources with the application of hybrid machine learning algorithms and MCDM methods to permit IoT applications to better adapt data processing and routing, providing a better experience for users. From the results, the proposed solution proved to be efficient and the adopted MCDM methods show a similar performance. This demonstrates the high level of flexibility in the ranking of alternatives for the proposed methods as a function of the adopted weights.

3.2 Chapter summary

This chapter presented the main related works, their contributions and limitations. The NSSF function is open field according to the 3GPP specifications, and several proposals have been presented. However, a set of issues make the slice selection problem challenging.

These questions involve:

Table 1 – Summary of Related Work

Work	Contribution	Limitations
Rivera et al. (2019) (RIVERA et al., 2019)	Traffic control rules can be deployed using minimal resources without influencing the efficiency of the Data Plane slices. A management system is necessary for the provision, updating, and control of the physical layer of VNFs that comprise the slices.	Lack of real world implementation
Bojkovic, Bakmaz and Bakmaz (2019) (BOJKOVIC; BAKMAZ; BAKMAZ, 2019)	SD-TOPSIS presents better performance in rank reversibility, but E-TOPSIS has significantly lower complexity operations. The E-TOPSIS method is suggested, especially when dealing with the fine granularity of slices. TOPSIS is considered a good decision-making tool, considering its algorithmic logic and mathematical form.	Lack of real world implementation. Fails to provide consistent results due to the rank reversal phenomenon.
Bakmaz, Bojkovic and Bakmaz (2020) (BAKMAZ; BOJKOVIC; BAKMAZ, 2020)	Alternative techniques, such as linear normalization (MAX-MIN), the weighting of variance, and binary classification alternatives, can reduce both the classification reversibility and computational complexity. This justifies the need to consider MCDM methods as a potential solution to the network slice selection problem. Lack of real-world implementation. To analyze the performance of other MCDM algorithms in terms of ranking reversibility.	Lack of real world implementation
Shurman, Rawashdeh and Jaradat (2020) (SHURMAN; RAWASHDEH; JARADAT, 2020)	A mechanism that enables user equipment to run multiple sessions on different network servers at the same time to utilize the advantages of their services.	Only a temporary session is allowed
Dimolitsas (2020) (DIMOLITSAS, 2020)	A multicriteria decision framework for the optimal selection of Edge Points of Presence (EPoPs) to deploy a network slice. Results indicate the relevance of the proposed two-stage method in meeting the user's hard and soft requirements, allowing communication between slices and optimal resource allocation from the providers.	High cost of deployment
Zhao et al. (2020) (ZHAO et al., 2020)	A Genetic Algorithm that can achieve satisfactory results in the maximization of user's Satisfaction Degree (SD) in the E2E network slicing problem. This method obtained better access and transmission performance when compared to traditional selection methods based on the Received Signal Strength (RSS) or greedy algorithms.	Lack of real world implementation
Otoshi et al. (2021) (OTOSHI et al., 2021)	A dynamic slice selection technique that learns to recognize the rough situation and the mapping between current situation and the future slice. The Bayesian Attractor Model (BAM) is used to achieve consistent recognition, as well as the Dirichlet Process Mixture Model (DPMM) to achieve automatic attractor construction. Situations mapping is also automatically learned by using feedback.	Problems such as the bit rate drop should be predicted in advance and the slices should be switched in advance
Silva et al. (2022) (SILVA et al., 2022a)	The use of hybrid machine learning algorithms and MCDM methods as a solution for the 5G network slice selection in IoT scenarios. The proposed solution proved to be efficient and the adopted MCDM methods show a similar performance.	Restrictions of the test environment

Source: elaborated by the author.

1. **Artificial Intelligence for IT Operations (AIOps)**: the use of artificial intelligence techniques requires re-training, adjustments of templates, configuration parameters, optimizers and models already trained during the network runtime, given the occurrence of abrupt changes in metrics that may or may not be associated to network anomalies, security issues, processing on nodes and/or need for scalability of cloud parameters and network elements. These points bring extra difficulties for the models proposal and to services orchestrations.

2. **Operational issues**: the slice selector must be multidomain if the local or regional orchestrator is multidomain. In this case, the architecture representation and the deployment models are linked to the telco orchestration service. In the event of a federated orchestration service: how is it possible to guarantee that the slice selection from user equipment is carried out end-to-end? What is the granularity of automation, MLOps, and interoperability required to orchestrate multiple elements across disparate administrative domains? The addressing of these points configures current research questions.

3. **Mathematical modeling**: the slice selection problem has stochastic features, dynamic, multi-variable, multiattribute, nonlinear, and time variant. Therefore, the proposition of optimal models are non-existent, given the difficulty of representing and predicting network QoS variables, thus making it difficult to use optimization methods.

The solution proposed in this work focuses on questions 1 and 2. The study's contributions are emphasized in Chapters 4 and 5.

4 PROPOSED SOLUTION

The slice-selection task at network runtime in a heterogeneous environment is a difficult problem, since there is still no fully accepted solution or technique in this field. This is because there are large number of variables and scenarios, as in the case of solutions that consider mobility between slices or not.

Thus, the implementation of new slice selection techniques becomes mandatory, including the demand for the growing use of vehicular networks, smart cities, robotics, agriculture 4.0, healthcare, remote surgery, UAVs, IoT, and IoV, among other technologies and scenarios involving network convergence.

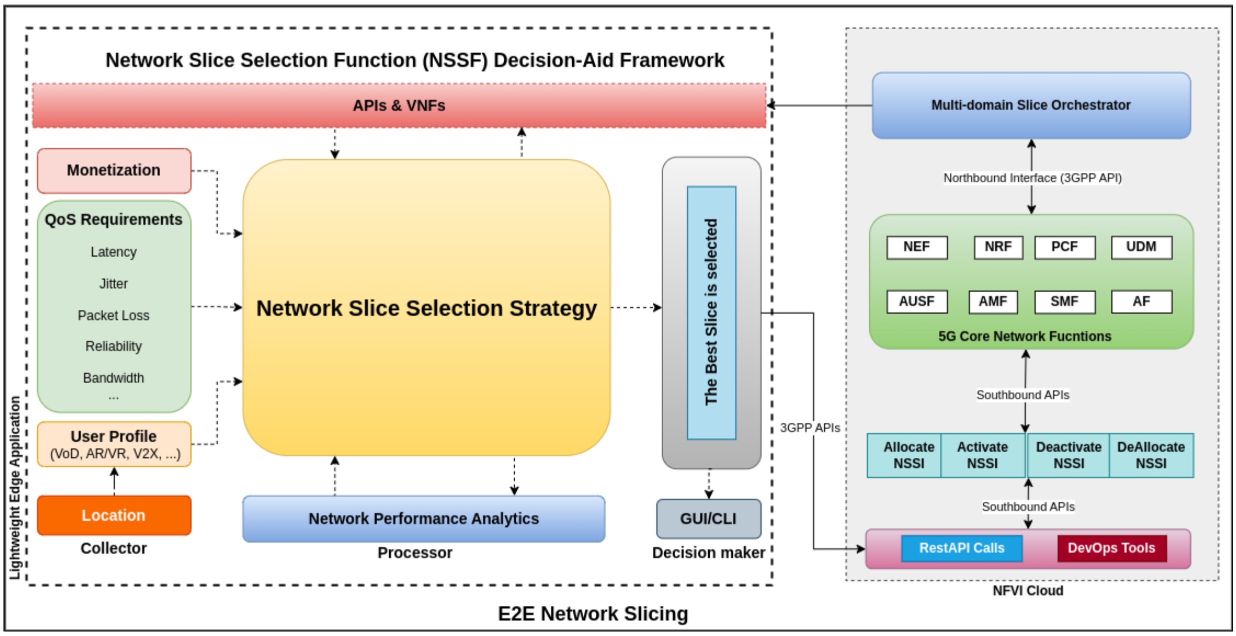
Overall, the literature suggests approaches that consider the following situation: given a set of criteria or network parameters, the system verifies at any time the available slices, which better meets the user needs, supporting the network exchange (handover process) for the mobile device (SILVA et al., 2022b),(SILVA et al., 2022a),(BOJKOVIC; BAKMAZ; BAKMAZ, 2019; SHURMAN; RAWASHDEH; JARADAT, 2020),(BARAKABITZE et al., 2020; TEAGUE; ABDEL-RAHMAN; MACKENZIE, 2019; KARATAS; KORPEOGLU, 2019).

This work presents a novel approach that employs several techniques aiming at integration and interoperability between RAN and core network through the proposed orchestration architecture, based on an efficient and robust NSSF that provides compatibility with standards specification (e.g. 3GPP, ETSI NFVI, and 5G PPP).

4.1 NSSF DAF: Network Slice Selection Function Decision-Aid Framework

Figure 4 provides an proposed framework overview for NSSF function. NSSF DAF consists of a solution that is partially executed on the user equipment (e.g. smartphones, vehicles, IoT brokers), running as a transparent service, while another one runs at the

Figure 4 – Proposed NSSF DAF: Network Slice Selection Function Decision-Aid Framework.



Source: elaborated by the author.

edge of the network operator or service provider. The framework has several modules that can be configured according to the context of applications, geographic location, scenarios of mobility, strategies of slices selection, among others. For energy saving, the user equipment only signals its consumption profile or user application preference to the framework hosted at the edge architecture. Hence, no processing occurs in the mobile device or in the IoT broker.

According to Figure 4, the framework proposed was divided into the following three main blocks: *Collector*, *Processor*, and *Decision Maker* of NSs. Fundamentally, the criteria for network selection are closely related to the demands or applications in use. Thus, parameters of application QoS, as well as objective quality metrics for specific applications, such as Quality of Video (QoV), and subjective metrics, such as indicators based on the user experience (QoE), must be considered (RICART-SANCHEZ et al., 2018; CHEN; ZHAO; LI, 2019). The *Collector* Module focuses on the assessment and mapping the appropriate QoS requirements dynamically for each type of service, in addition to considering the signaling User Profile realized by the mobile device to different scenarios (e.g. V2X, VR, AR, Video on Demand (VoD), Video Stream). Moreover, other variables can be considered, such as monetization and geographic location (RICART-SANCHEZ et al., 2018; AFAQ et al., 2020; LU et al., 2020).

NSSF DAF is indifferent to the technique used to mark packages in gNodeB. The architecture proposed here assumes that a software instance in Centralized Unit (CU) and Distributed Unit (DU) based on widely used solutions, such as Segment Router (SR), SDN Flow Tables, EVPN (VPN Ethernet), VxLAN (Virtual eXtensible Local Area Network), Multi-Protocol Label Switching (MPLS), and/or definitions of IPv6 classes of service have already marked (labeled) the packages (3GPP, 2020). Therefore, the framework only identifies and collects the QoS and QoE parameters. The data are processed them, thence, defines which NS best meets those requirements; or, in case of non-existence, it signals to the Orchestrator the parameters for instancing the slice at run time (already in the cloud).

The Module *Processor* uses preferentially models that consider hybrid solutions: MCDM methods, fuzzy logic, and machine learning; as mentioned in Section 2.3. In general, these models work similarly, that is, after the data collection process, the slices are classified and ranked; finally, the selection via the decision strategy is performed. In this case, each criterion receives a certain weight to prioritize some services over others, guiding the new slice choice according to the application in use. The definition of weights also takes into account the user profile and other aspects of the available slices. As described in (SILVA et al., 2022b) and (BONATI et al., 2020), all computational methods are implemented using VNFs, running in virtualization solutions, for example, on OpenStack (OPENSTACK, 2022) and Kubernetes (K8s) (KUBERNETES, 2022).

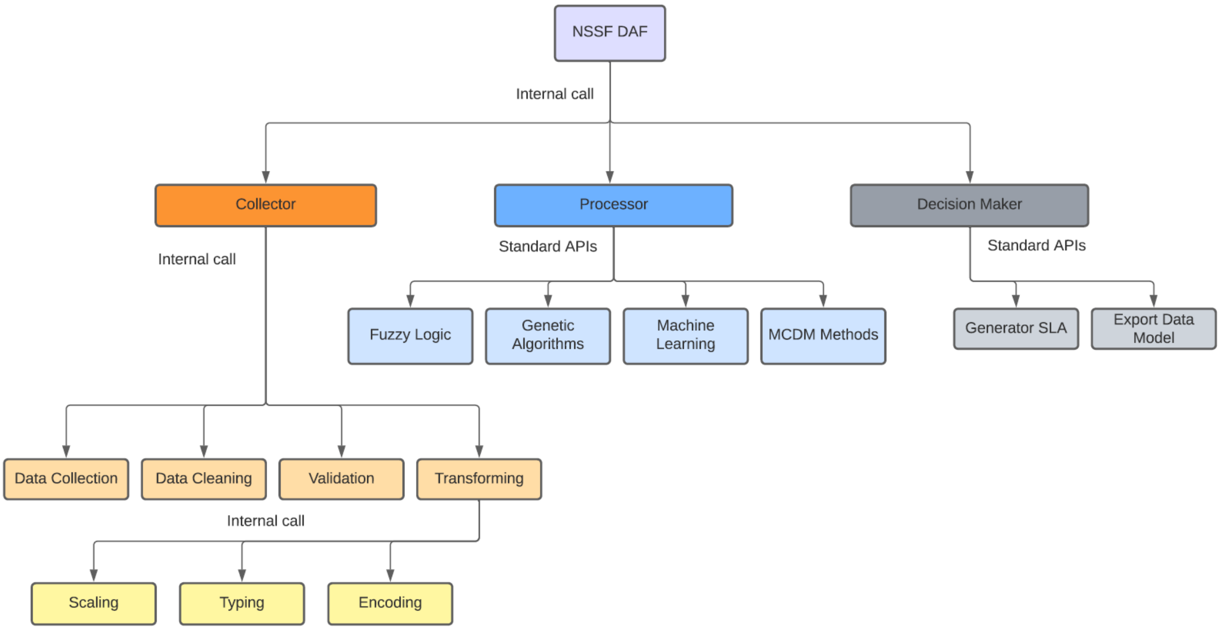
For data collected, as well as the results persistence related to the processing module, a database is used. Data can be manipulated and analyzed by computational intelligence algorithms with the use of APIs, consuming data directly from the *Network Performance Analytics* module, environments based on Hadoop (HADOOP, 2022) and Spark (SPARK, 2022) ecosystems. Note that the framework supports any relational and non-relational database.

The *Decision Maker* Module selects the slice that best fits the UE data stream and then aggregates the network traffic and redirects it to the slice that meets the required SLA configuration. All the operations between modules and blocks of the framework are performed via APIs, which provide external access by other entities through RSA[1] key pairs (Secure Shell (SSH) keys) (PAVANI; SRIRAMYA, 2021).

Information about traffic conditions on slices, as well as extra structure configuration options, are available on a panel in the Graphical User Interface (GUI) or via Command Line Interface (CLI) when accessing the slice selection service. It is possible to combine

[1]This algorithm is called RSA because of surnames of the three authors (Ron Rivest, Adi Shamir, and Leonard Adleman).

Figure 5 – NSSF DAF Structure specification.



Source: elaborated by the author.

more than one selection strategy or method, for example, using machine learning algorithms, fuzzy logic, or multi-criteria decision methods. It is also possible to define which QoS parameters should be considered in the selection process.

4.1.1 Structure specification

The NSSF DAF is structured around three large blocks or main modules, as shown in Figure 5.

The *Collector* module presents a set of components responsible for data acquisition and pre-processing. The characteristics and functionalities of these components are detailed below:

1. *Data Collection*: the component presents a set of methods responsible for obtaining the QoS metrics (e.g. Latency, Jitter, Packet Loss, Reliability, Bandwidth), QoE, and user consumption profile (e.g. VoD 4K, AR/VR, V2X). This component uses passive measurement techniques, using packet analyzers and protocols (sniffers). It can be configured to use active measurements through Internet Control Message Protocol (ICMP) requests, collecting information, such as Round-Trip Time (RTT), among other metrics. The module offers data consumption routines through publish/subscribe solutions.

2. *Data Cleaning*: this component is responsible for data cleaning, the fixing of reading errors, and dataset structuring.

3. *Validation*: the component is responsible for verifying whether the set of metrics received satisfies what is expected from a dataset.

4. *Transforming*: the Transforming component has the following three subcomponents: Scaling, Typing, and Encoding. The Scaling subcomponent performs the normalization process with libraries and methods, which facilitates the application of machine learning techniques. The Typing subcomponent makes the necessary data typing conversions (e.g. float to integer, integer to float, string to float). Finally, the Encoding subcomponent performs the necessary transformations in categorical metrics by converting them to numeric classes, in order to include them in the machine learning models.

The *Processor* module presents a set of components prepared to use several optimization techniques, computational intelligence, stochastic and multi-criteria decision approaches. The components' characteristics and functionalities are detailed as follows:

1. Fuzzy Logic: the component supports the use of fuzzy logic, its membership functions, fuzzification, and defuzzification methods. The *Collector* module output allows for component integration transparently.

2. Genetic Algorithms: the component allows for the use of genetic algorithms. The *Collector* module organizes the data, thereby facilitating integration, the definition of generations, and an initial population. It also facilitates the application of multi-objective formulations directly to the dataset.

3. Machine Learning: the component supports the use of machine learning techniques. Data organization performed by the *Collector* module facilitates the use of supervised models. It also enables reading non-relational databases, batch files, or data streaming, which also allows for the use of unsupervised models.

4. MCDM Methods: the component allows for the use of several multi-criteria decision methods. The structuring of datasets facilitates the assembly of decision matrices. The criteria weights can be defined directly by the network operator or service provider, and can also be obtained by using hybrid approaches with machine learning techniques. Weights can even be mapped directly from the end user's consumption profile. These integrations take place through APIs.

The *Decision Maker* module presents the components that are necessary to integrate the techniques used in the *Processor* module, with Cloud Computing tools and solutions, especially the network orchestrators. The characteristics and functionalities of these components are detailed below:

1. *Generator SLA*: the component is responsible for generating the SLA that will be used in the creation of on-demand slices (horizontal model), or for forwarding data flows to the already available and pre-configured slices (vertical model). The SLA is assembled from the output of the *Processor* module and fulfils the requirements of the application in use or the user's consumption profile.

2. *Export Data Model*: this component uses multiple data models that facilitate integration with third party tools, especially network orchestrators. The transport of data models takes place via the consumption of APIs.

4.1.2 Integration overview

Figure 6 presents an overview of the proposed solution integration in the context of 5G/B5G networks. Initially, the NSSF DAF had been designed to function as an MEC (Mobile Edge Computing) service, being one of the main contributions of this work. However, the objective was to evaluate the implementation in a local datacenter close to the gNodeB in the RAN, or in the 5GC. It seeks to verify whether there is a significant difference in the performance of slice selection at different points of the network slicing architecture. Its integration with the data plane (UPF—User Plane Function) is essential for traffic recognition and the differentiation of service classes, which enables its operation.

To enable the definition of requirements, the limitations of the works presented in Chapter Y were considered. Among the main functional and non-functional requirements (R) of the NSSF DAF, we identified the following:

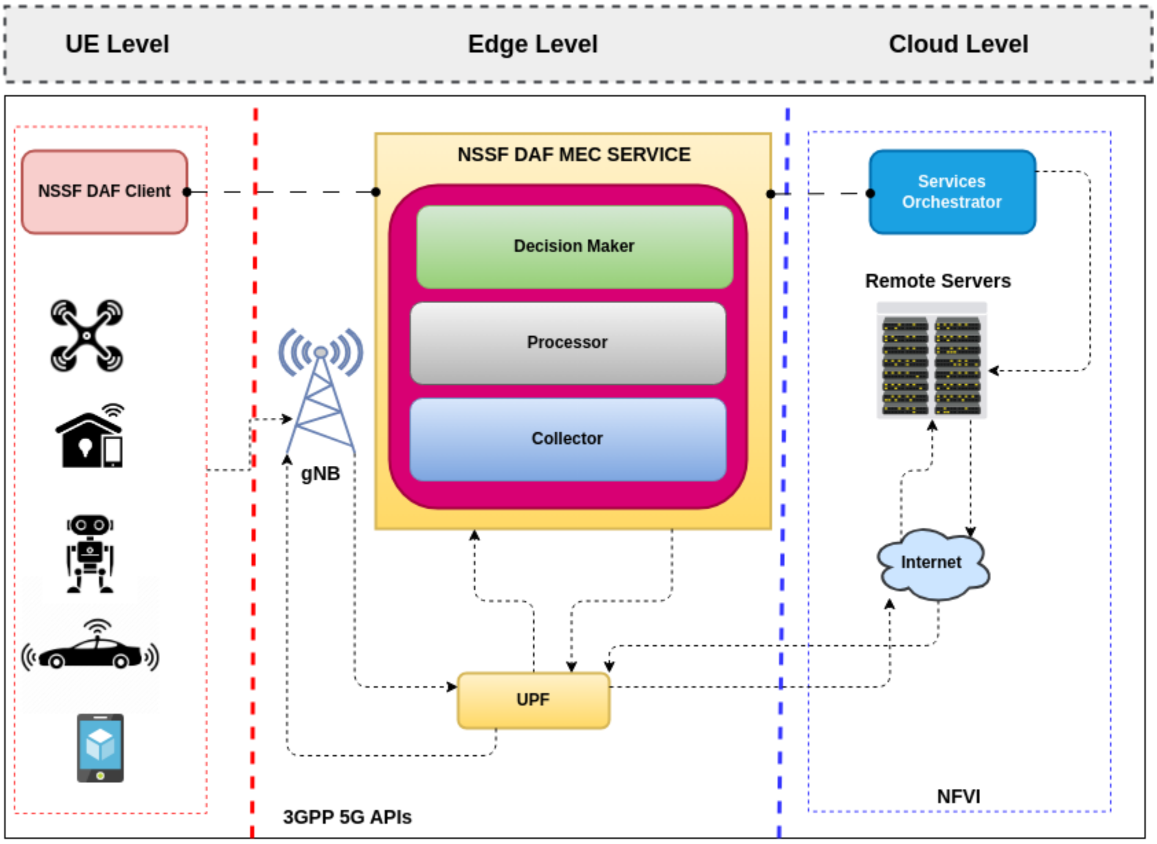
$R1$: Deployable on servers with different virtualization solutions (NFVI–NFV Infrastructure) and cloud orchestration platforms;

$R2$: Facilitate deployment in production networks, in a transparent way;

$R3$: Facilitate the integration of later modules, adding new features, through the available documentation;

$R4$: Compliance with the specifications and models of the standards bodies (e.g. 3GPP and ETSI);

Figure 6 – NSSF DAF Integration overview.



Source: elaborated by the author.

*R*5: Act independently of specific hardware manufacturers and models;

*R*6: Use open-source solutions;

*R*7: Use DevOps—Development and Operations—philosophy, practices, and tools; MLOps—Machine Learning Operations; AIOps—Artificial intelligence for IT operations;

*R*8: Make the mobile multiplatform application available for download from public repositories.

The NSSF DAF - Client application is a cross-platform and can be embedded in several UEs, such as UAVs, vehicles, smartphones, robots, or even in IoT brokers in smart homes. The application uses a set of APIs, providing the necessary signaling for the NSSF DAF - Server. The UE sends messages with its location (GPS coordinates), checking whether there is an occurrence of NSSF DAF on the Edge or RAN to which it is linked. From this signaling and the profile of the application in use, the NSSF DAF selects the best slice available, and returns with the best option for the UE, as detailed in the modules included in Section 4.1.3 and illustrated in Figure 8.

The functional and non-functional requirements (RA—Requirement Application) of the NSSF DAF - Client application are detailed below.

*RA*1: Become able to search for the available slices (RAN and Edge);

*RA*2: Direct a request for the best slice option to NSSF DAF, through HTTP requests and API consumption;

*RA*3: Execute the handover to the best selected slice;

*RA*4: Facilitate the setting of the user's profile (preferences), which can be as follows: VoD 4K/8K, AR/VR, V2X, among others;

*RA*5: Run in automatic (service) or manual mode, according to the UE.

Regarding integration with cloud computing tools, the NSSF DAF operates with standardized data models, allowing for the required integration. The specification of the functionalities of all modules is detailed in Section 4.1.1 and the implementation model in Section 4.1.3.

4.1.3 Implementation model

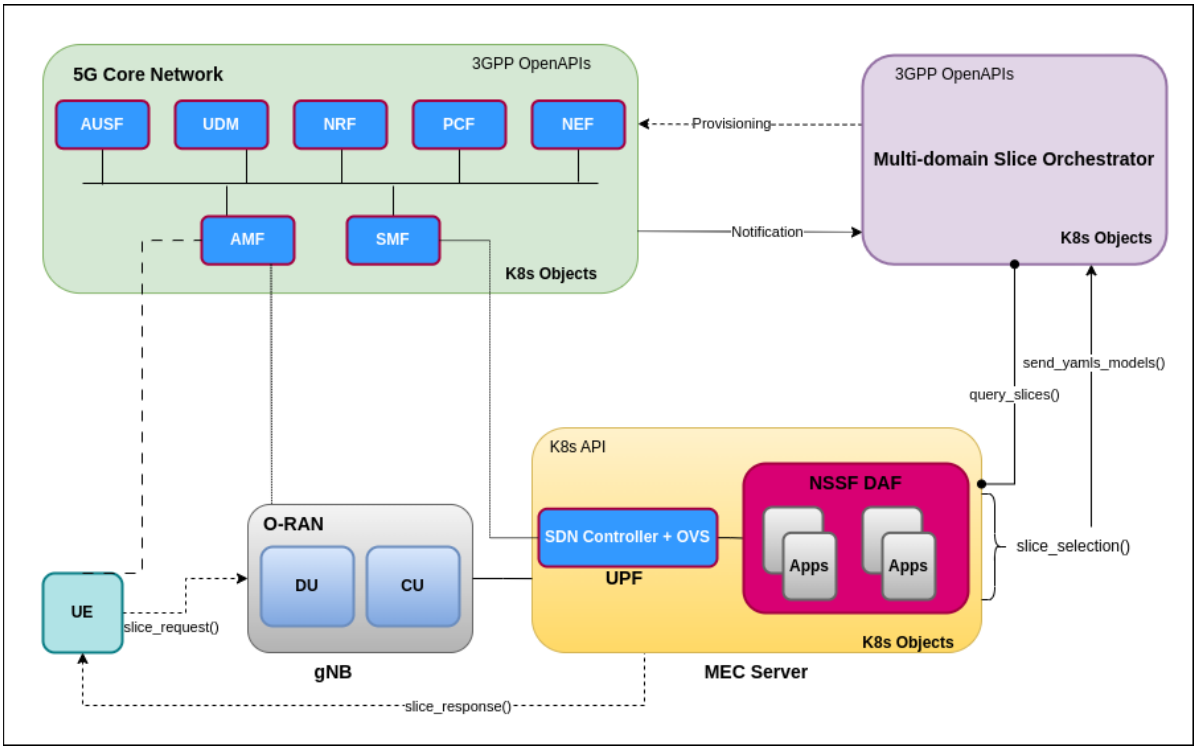
Overall, the implementation model comprises the hierarchy as well as the technical representation of how the system works. For this, there are diagrams descriptions, the subsystems implementation, and the representation of the interfaces, services, pipelines, and components. In addition, supplemental information on the structure and management of the proposed slice selection function provided, based on the details of some technologies that make up the NSSF DAF framework.

The purpose of this section is the technical specification detailed of our solution, as described in Section 4.1.1. This process is segmented into two parts: i) detailing the framework's general architecture from the technologies used to consolidate the models and implemented techniques; ii) validation and analysis of the proposed algorithms through experiments in testbed. The configuration of the different scenarios allow for the verification and validation of the entire proposed solution, and are detailed in Chapter 5.

Figure 7 illustrates different implementation blocks. For the NFs of 5G Core Network, the deployment is carried out in different PODs[2]. Each POD can contain one or more

[2]Pods are the smallest deployable units of computing used in K8s cluster. For more information see https://kubernetes.io/docs/concepts/workloads/pods/.

Figure 7 – NSSF DAF Implementation model.



Source: elaborated by the author.

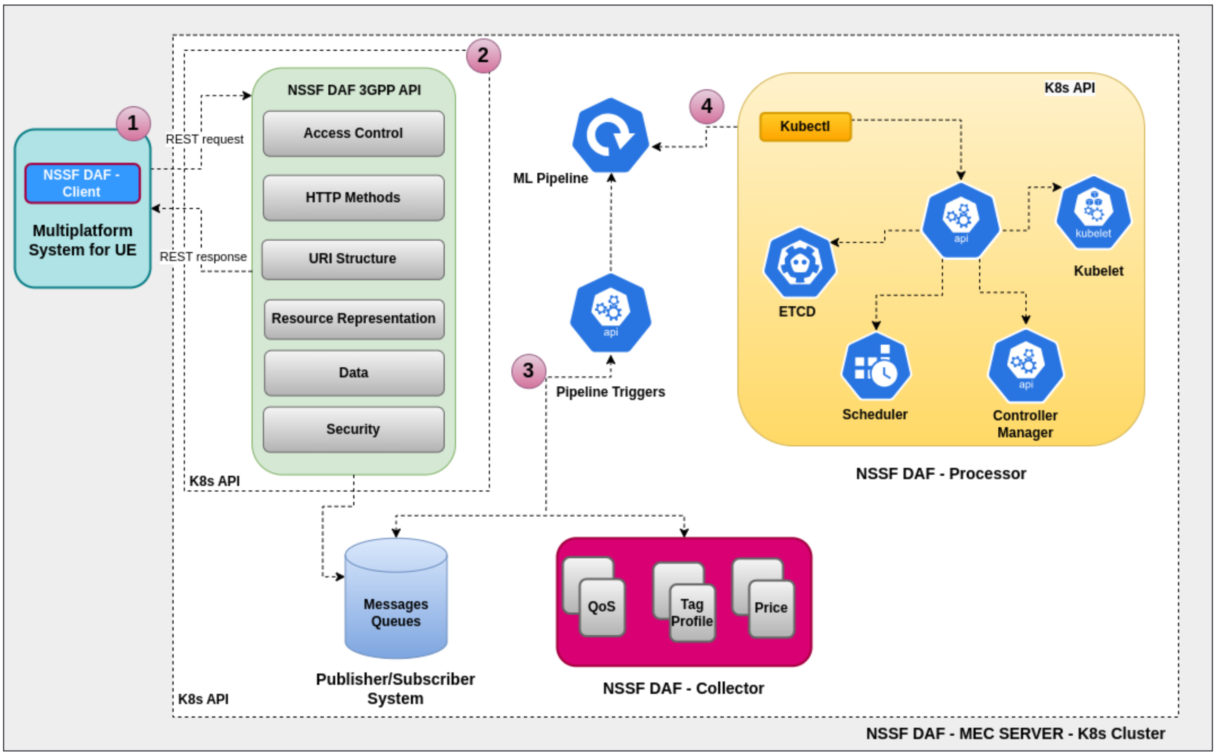
containers, this sizing is done based on network traffic and depends on the scalability model and balancing of computational resources of the K8s cluster. The UE can be any mobile device with access to the 5G network, or even an IoT broker or CPE (Customer Premise Equipment). The connection between the access network and the edge cloud occurs through VNFs that use SDN (protocols and controllers) and NFV (OpenvSwitch) technologies.

The messages exchange between the architecture functional blocks and service requests are performed through the consumption of APIs. The objects and components that form the implementation model are K8s objects, and, therefore, follow the service lifecycle management and features inherent to the CNFs. In addition, all provisioning is done from the concept of infrastructure as code, and uses tools such as Ansible[3], Chef[4], Puppet[5], as well as data models such as YAML (YAML Ain't Markup Language), XML (Extensible Markup Language) and JSON (JavaScript Object Notation).

Figure 8 shows the integration from the last mile to the edge, according to the perspective of the client application. Note that four main steps are presented. Each of these

[3]https://www.ansible.com/
[4]https://www.chef.io/
[5]https://www.puppet.com/

Figure 8 – NSSF DAF - Client : Operations and integration with edge computing architecture.



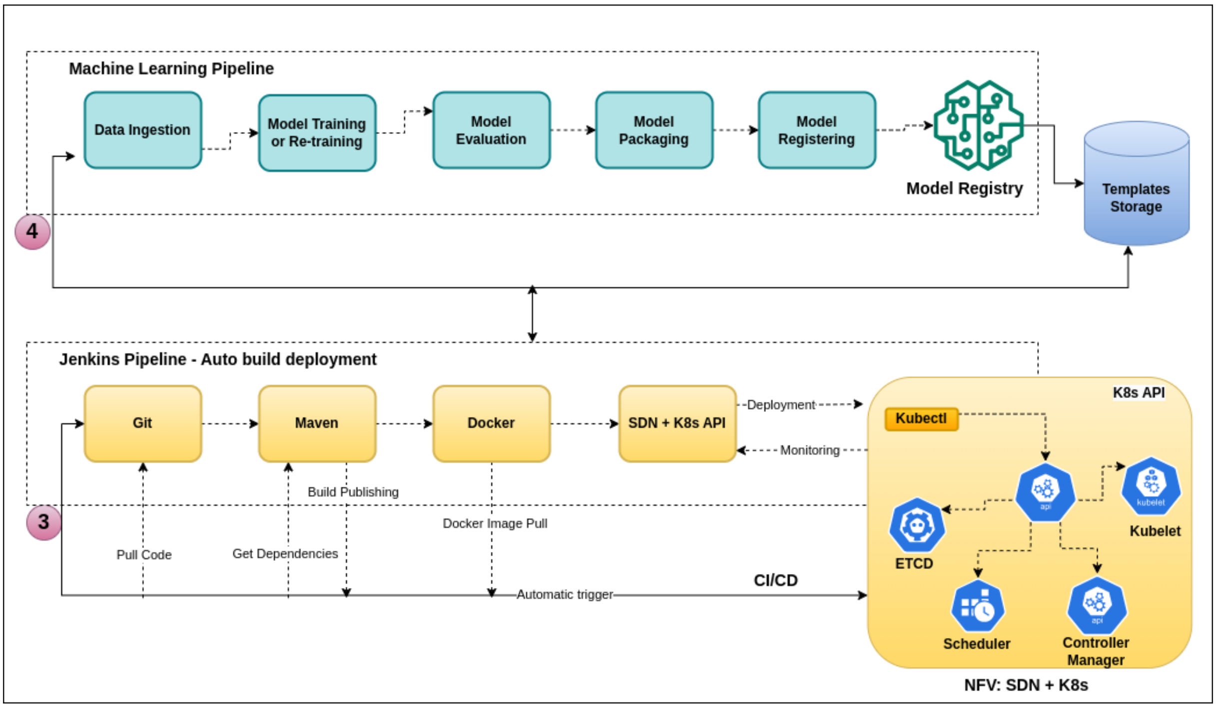
Source: elaborated by the author.

steps form different functional blocks and are implemented by a CI/CD (Continuous Integration/Continuous Delivery) process. The first step consists of implementing the RAs of our solution defined for the client side, NSSF DAF - Client, as described in Section 4.1.2. The multiplatform application demands services to the edge cloud, in this case, it requests for available slices, passing your consumer profile as a parameter. In the second step, a set of APIs are dimensioned to provide different services, across a RESTful implementation.

The second step illustrated in Figure 8 also shows the NSSF DAF 3GPP API block, which follows the 3GPP TS 29.501[6] specification. This block comprises the use of HTTP methods, which follow compliance standards, and allow for their integration with different web technologies, supporting different cryptographic algorithms and data representation. Furthermore, it enables the use of different data modeling languages (e.g. YANG, NETCONF and RESTCONF). Note that this stage also supports the use of publisher/subscriber systems, which, in this framework, can be implemented with different tools; for example, the use of Apache Kafka[7] and RabbitMQ[8] servers. These tools are crucial in 5G and future network scenarios due to the density of service requests, especially

[6]https://www.3gpp.org/DynaReport/29501.htm
[7]https://kafka.apache.org/
[8]https://www.rabbitmq.com/

Figure 9 – NSSF DAF using MLOps pipeline and Cloud-Native Network Functions.



Source: elaborated by the author, based on (YAN; SHENG, 2023) and (KREUZBERGER; KüHL; HIRSCHL, 2023).

in eMBB and mMTC scenarios.

Figure 9 describe the implementation blocks comprising the third and fourth steps. In the third step, its shows a high-level description of a pipeline used in cloud-native solutions. In this context, different tools can be used to implement these pipelines. In the solution proposed in this work, Jenkins was used in collaboration with the Ansible tool (R; MOHANA, 2022). Both tools allow for the construction and automation of MLOps workflows, as well as they provide efficient implementation of CI/CD components (e.g. Build, test and push operations). The triggers can occur sequentially or in parallel and allow interoperability between different tools, such as Git, Maven (Pymaven), Docker, SDN Controllers and K8s Cluster (Master and Nodes). These tools provide specific services and collaborate for the complete operation of the solution. Each of these applications can run on different PODs, and are interconnected through a Calico network (SDN for Containers).

Finally, in the fourth step, there is the ML model deployment. Initially, the feature engineering pipeline is performed by the Collector module of the NSSF DAF framework, described in Section 4.1.1. After the data acquisition, cleaning, transformation and validation steps, the automated ML workflow pipeline starts. This phase is divided into the following steps: i) Data Ingestion; ii) Model Training and retraining; iii) Model Evaluation;

iv) Model Packaging; and v) Model Registering (KREUZBERGER; KüHL; HIRSCHL, 2023).

Two phases ii) and iii) in the ML pipeline deployment process should be highlighted. In the training and retraining phase of the models, different operations are performed to adjust the hyperparameters, optimizers, and activation functions in order to enable performance gains of the techniques used; in the evaluation phase, the models are tested on a set of test data aiming at the validation of its performance. The final phases consist of the deployment of tested models in CNFs. All models are stored in a repository in the form of templates, which can be loaded quickly depending on the traffic conditions in the slices, and the demand for adjusting metrics in network and cloud orchestrators to guarantee the SLAs (YAN; SHENG, 2023; BELTRE; SAHA; GOVINDARAJU, 2021; RAJ et al., 2021).

The second part of validating the implementation model of NSSF DAF framework is detailed in Chapter 5.

4.2 System scalability and extensibility

The proposed framework is scalable and can be used with new criteria for slice selection, as well as different access network technologies. In addition, the solution deployment model can be adjusted according to the network operator's preferences or service provider's characteristics, as mentioned earlier in this text. In this sense, the dynamism provided by cloud orchestration tools, such as Kubernetes or OpenShift[9], is used. It is possible, therefore, to have numerous instances of the NSSF DAF framework at different points in the network architecture, scaling the solution according to the scenario.

The interaction between NSSF DAF instances, even in different administrative domains, provides about the quality of available slices in certain regions (e.g. neighborhoods, universities, cities). NSSF DAF also allows for the addition of new modules, for example, a broker service for IoT devices that support Message Queuing Telemetry Transport (MQTT). It can also facilitate the attachment of a Mobility Management (MM) module, selection of Access Points (APs) for RAN based on Wi-Fi 6.

The framework deployment can be performed by using DevOps and MLOps tools, and it can still be deployed in containers or virtual machines on a public cloud or local data center, according to the details about the implementation model in Section 4.1.3.

[9]https://openshift.com/

This feature can be used on private and public 5G networks.

4.3 NSSF DAF restrictions

Although the NSSF DAF framework solves the slice selection problem, it does not address issues related to MM. Thus, the mobile device must also use a management protocol that allows for handover between slices, keeping the continuity of the service active, especially in instances sensitive to latency and packet loss, such as traditional real-time applications.

In (JAIN; LOPEZ-AGUILERA; DEMIRKOL, 2020) the main challenges involving MM are presented, as well as potential solutions. Among the main issues, we can mention handover signaling, network slicing, integration with other frameworks, and frequent handovers, among others. Potential solutions include the use of deep learning, the creation of mechanisms on demand MM, SDN- NFV integration with legacy methods (e.g. IEEE 802.21, PMIPv6, LTE MM) and the use of edge computing solutions.

These demands are outside the scope of this work, but the framework proposed here is in line with these demands, and can be easily extended to cover these issues. This is possible due to its deployment through the edge computing model, SDN and NFV, in addition to slice selection strategies, Radio Access Technologies (RAT), the network metrics acquisition approach and slice evaluation with traffic flow during operation.

4.4 Chapter summary

This chapter presents the most relevant contribution of this work. Overall, a framework for the slice selection problem was proposed, in accordance with the main specifications of 3GPP and ETSI. The solution enables its application in real environments and in networks that are already in production, facilitating its integration into the Non-Standalone Architecture (NSA) and Standalone Architecture (SA) 5G models and future networks.

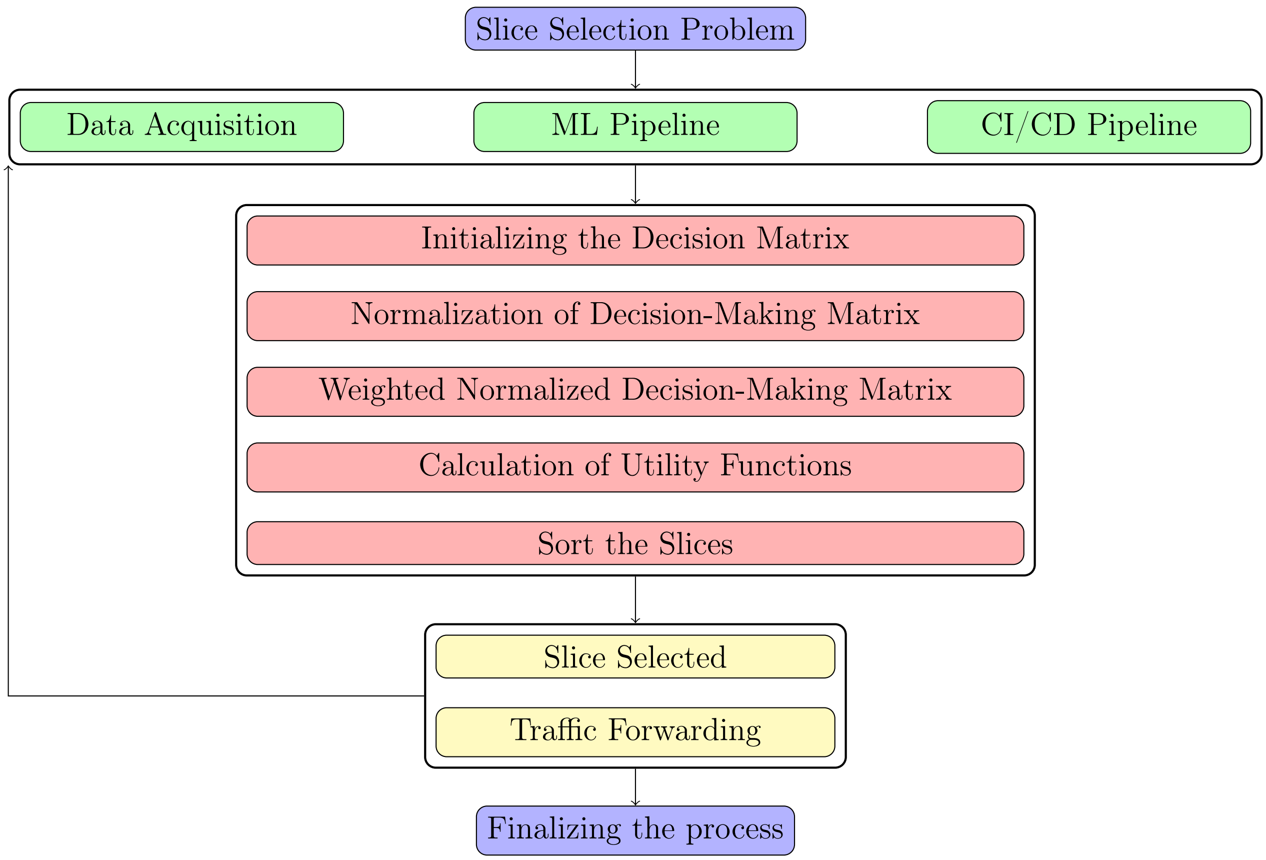
The technical details of the modules, the hierarchical structure of the solution, the techniques and methods involved are presented; also, there is the definition of the functional and non-functional requirements of the edge application and at the user level. The integration model of the solution is also detailed, and its adequacy to architectures both under development and already standardized in the literature. It is important to emphasize that all requirements have been implemented, verified and validated.

In order to validate the proposed solution, the next chapter is dedicated to the detailing of the tests performed and the description of the evaluated scenarios. The objective is to measure the robustness, efficiency and practicality of the solution.

5 NSSF DAF PROOF-OF-CONCEPT TEST-BED

In order to validate the proposed solution, we conducted experiments integrating an approach that uses ML and decision-making methods to support NSSF DAF operations in multi-domain slicing environments. The workflow shown in Figure 10 outlines the deployment process for proposed solution in this work. It's important to note that the slice selection problem involves several components and tools. The details solution are discussed in Sections 4.1.1 and 4.1.3 in Chapter 4, including component organization and modules, and are visually represented in Figures 7, 8, and 9.

Figure 10 – NSSF DAF workflow.



Source: elaborated by the author.

The methods proposed in this approach focus on the evaluation and the dynamic mapping of adequate QoS requirements. By the analysis of network traffic coming from the UEs, a template to create slices in various formats is composed (e.g. YAML, NETCONF, and JSON), allowing for their integration with 5G Core network functions, and, therefore, the use of coding, automation, and network service orchestration tools, such as Ansible, Chef, and Puppet. The NSSF DAF framework also facilitates integration with ONAP[1] Istanbul/Honolulu and OSM[2] 12 orchestrators.

5.1 NSSF DAF: Network Slice Selection Strategy

The NSSF DAF framework processing module was developed by implementing a set of MCDM methods in Python[3] language version 3.8. Thus, the Aras, Cocoso, Codas, Copras, Edas, Mabac, Mairca, Marcos, Moora, Ocra, Promethee II, Spotis, Topsis and Vikor methods were implemented. The initial choice of these methods was made due to their wide use in the literature and proved robustness, allowing proposed solution modules. However, other strategies to process data from the *Collector* module can be explored.

Algorithm 1 is responsible for slice selection, constituting the most important part (core) of the proposed framework. Initially, we used the function *ParserStream2DataFrame*, which is implemented in the Algorithm 2. This function aims to obtain the dataset that is used in decision matrices, constituting the first step of the processing MCDM methods. The QoS and QoE parameters are concatenated, verified, and validated. The dataset produced from this procedure serves as the input for decision-making methods. Figure 10 shows this workflow.

The *applyMethods* function (see Algorithm 1) receives the original dataframe of the *Collector* module, which can be in a data stream or batch file. The QoS parameter is either of the type "the bigger the better", which is represented in the array as 1, or "the smaller the better", which is defined in the array as -1. Lines 5 to 18 of the algorithm correspond to the application of MCDM methods on the same dataset. These tasks are presented sequentially for better visualization, but they are executed in parallel.

Furthermore, the algorithm allows the extra parameters of the implemented decision methods. For example, for the Promethee II method, the usual criterion curve was considered because the parameters dynamics and the network traffic conditions change sharply in short periods. The default configuration was utilized for the remaining methods. A

[1]https://www.onap.org/
[2]https://osm.etsi.org/
[3]https://www.python.org/

Algorithm 1: NSSF Decision-Aid Framework.

Input: Number of evaluated slices N, Number of QoS/QoE Criteria C, Dataframe with slice data $DF(r, c)$, array of weigthts W, Criteria types T, MCDM methods M, Methods parameters P

Output: The Best Slice is selected S, slice ranking $R(i)$, SLA slice template F

begin
- $DF \leftarrow$ ParserStream2DataFrame(N, C) ; /* apply Algorithm 2 */
- **while** $DF \neq \emptyset$ **do**
  - $DFR \leftarrow$ applyMethods (DF, M, W, T, P) ;
    /* making preferences ranking by method */
  - $RA \leftarrow$ rankDataMethod (DFR) ; /* Aras ranking */
  - $RCO \leftarrow$ rankDataMethod (DFR) ; /* Cocoso ranking */
  - $RC \leftarrow$ rankDataMethod (DFR) ; /* Codas ranking */
  - $RCP \leftarrow$ rankDataMethod (DFR) ; /* Copras ranking */
  - $RE \leftarrow$ rankDataMethod (DFR) ; /* Edas ranking */
  - $RM \leftarrow$ rankDataMethod (DFR) ; /* Mabac ranking */
  - $RMI \leftarrow$ rankDataMethod (DFR) ; /* Mairca ranking */
  - $RMC \leftarrow$ rankDataMethod (DFR) ; /* Marcos ranking */
  - $RMO \leftarrow$ rankDataMethod (DFR) ; /* Moora ranking */
  - $RO \leftarrow$ rankDataMethod (DFR) ; /* Ocra ranking */
  - $RP \leftarrow$ rankDataMethod (DFR) ; /* PrometheeII ranking */
  - $RS \leftarrow$ rankDataMethod (DFR) ; /* Spotis ranking */
  - $RT \leftarrow$ rankDataMethod (DFR) ; /* Topsis ranking */
  - $RV \leftarrow$ rankDataMethod (DFR) ; /* Vikor ranking */
  - $S \leftarrow$ sort$(RA[0], RCO[0], RC[0], RCP[0], RE[0], RM[0], RMI[0], RMC[0]$, RMO[0], RO[0],RP[0],RS[0],RT[0],RV[0]);
  - $R \leftarrow$ sort$(RA, RCO, RC, RCP, RE, RM, RMI, RMC, RMO, RO, RP$, RS,RT,RV);
  - $SLA \leftarrow$ saveResult $(RA, RCO, RC, RCP, RE, RM, RMI, RMC, RMO$, RO, RP,RS,RT,RV); /* data for 3GPP API REST */
  - $F \leftarrow$ exportTemplateSLA (SLA, S, R) ;
- **end**
- **return** F;

end

Algorithm 2: Parser Stream to DataFrame

Input: (S, P), where S is the stream or log of the network flow; P the type settings to convert.

Output: DataFrame d

ParserStream2DataFrame `ParserStream2DataFrame(`S, P`):`
- $dataQoS \longleftarrow ParserMainParams(S, P)$
- $dataQoE \longleftarrow ParserExtraParams(S, P)$
- $d \longleftarrow concatenate(dataQoS, dataQoE)$
- $d \longleftarrow validateAndSetTypes(d)$
- **return** d

clear example of the Algorithm 1 execution can be consulted in Appendix A (more details in (NGUYEN et al., 2022a; WANG et al., 2020; ULUTAş et al., 2020; DEZERT et al., 2020; KUNDAKCı, 2019; YAZDANI et al., 2019; ALINEZHAD; KHALILI, 2019; WĄTRÓBSKI et al., 2019; HEZER; GELMEZ; ÖZCEYLAN, 2021; BOJKOVIC; BAKMAZ; BAKMAZ, 2019; BABASHAMSI et al., 2016; FIGUEIRA; GRECO; EHRGOTT, 2005)).

The *rankDataMethod* function defines best slices ranking for each method evaluated. This method implements the *Processor* module in the multi-criteria decision strategy (MCDM Methods), described in Section 4.1.1. The other functions (*saveResult* and *exportTemplateSLA*) presented in the algorithm are responsible for handling the resulting data and configuring the SLA template in a 3GPP REST format. These functions facilitate integration with 5G CORE and, therefore, with orchestration tools. All computational methods are implemented using VNFs, running in an environment based on K8s Cluster (YAN; SHENG, 2023), (BELTRE; SAHA; GOVINDARAJU, 2021), (BONATI et al., 2020).

5.2 Environment description

To explore, validate, and develop ML models, a network traffic dataset is required. This work followed the standard pattern presented in several articles by building the scenario and producing a dataset based on a testbed. This is necessary due to computer network scenarios' wide and diverse characteristics.

5.2.1 Test Setup 01

The test environment was implemented using the network software emulator GNS3[4] (Graphical Network Simulator 3) to provide the infrastructure for the virtualized network functions (NFV Infrastructure). The scenario is composed of 05 UEs transmitting at different traffic rates using a client Iperf[5] and Scapy[6] librarie, as specified in Table 2, for collections set performed. In addition, implementation in the gNodeB (3GPP 5G Next Generation base station) was performed through a software module (CUPS—Control User Plane Split) running on an Ubuntu[7] Linux 16.04 LTS VM with Kubernetes, based on the

[4]https://www.gns3.com/
[5]https://iperf.fr/
[6]https://scapy.readthedocs.io/en/latest/usage.html
[7]https://ubuntu.com/

Table 2 – Specification of injected traffic during 5 min for each collection.

Collection	**UE 01** (Mbps)	**UE 02** (Mbps)	**UE 03** (Mbps)	**UE 04** (Mbps)	**UE 05** (Mbps)
01	20	47	07	09	06
02	22	45	06	12	08
03	24	49	09	10	06
04	32	55	14	18	12
05	25	50	10	05	02
06	45	70	30	05	03
07	12	25	05	20	30
08	20	22	24	02	07
09	08	28	05	10	03
10	32	62	22	15	08
11	10	15	05	02	15

Source: elaborated by the author.

OpenAirInterface (openair-k8s) project (OPENAIRINTERFACE, 2022).

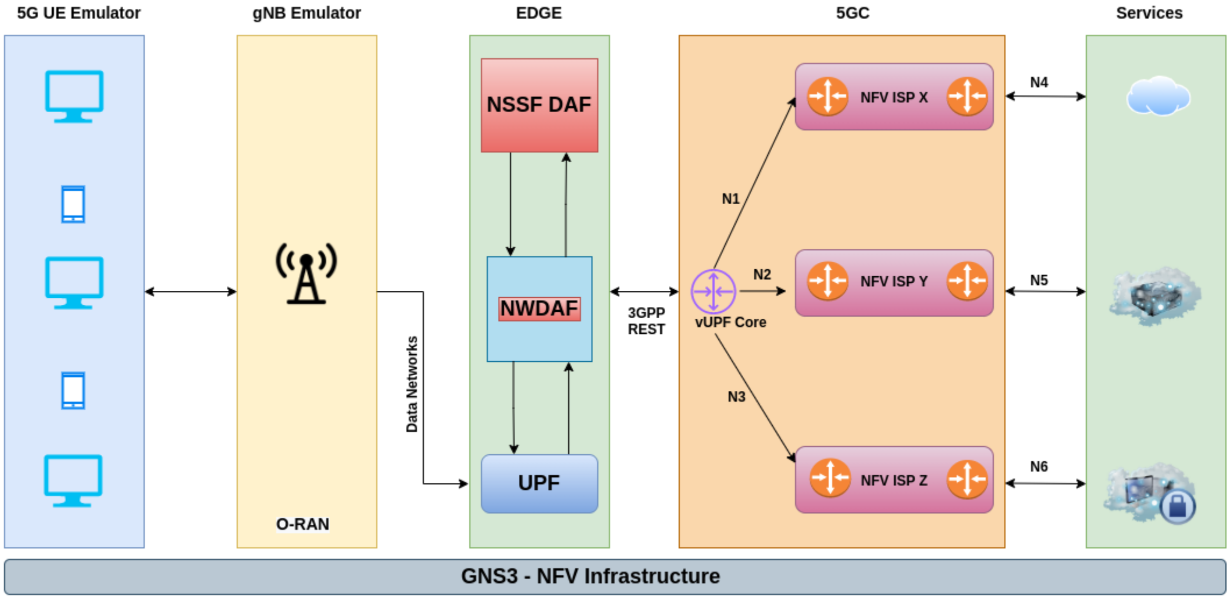
Regarding the network edge (Edge Computing), a docker[8] container was used with the deployment of SDN controller OpenDayLight[9], to implement the UPF, responsible for data flow separation and identification. The Network Data Analytics Function (NWDAF) was implemented in an Ubuntu Linux 16.04 LTS VM with the Anaconda[10] Python 3 framework, to implement the analysis functions and the data mining pipeline, using the Jupyter[11] notebook tool and the Numpy[12], Pandas[13] and Sklearn[14] libraries.

Although the scenario presented the 5G architecture core (5G Core—5GC), any Network Functions (NFs) relevant to the core were implemented, such as Network Exposure Function (NEF), Network Repository Function (NRF), Policy Control Function (PCF), Unified Data Management (UDM), Authentication Server Function (AUSF), Access and Mobility Management Function (AMF), Session Management Function (SMF) and Application Function (AF), as shown in Figure 11. These network functions will be implemented through ETSI NFV OSM, which will use the SLA template generated by the NSSF DAF framework.

However, as detailed below, three vertical slices were considered delivered at the network edge by three different 5G providers, according to the 3GPP TS 22.186 V16.2.0 and

[8]https://www.docker.com/
[9]https://www.opendaylight.org/
[10]https://www.anaconda.com/
[11]https://jupyter.org/
[12]https://numpy.org/
[13]https://pandas.pydata.org/
[14]https://scikit-learn.org/stable/

Figure 11 – Illustration of the simulation environment architecture for producing network traffic.



Source: elaborated by the author.

3GPP TS 22.261 V16.14.0 specifications (ETSI, 2020),(ETSI, 2021). Each composition slice is given by the junction of Virtual Networks (VNs)' $N1$ to $N6$, under three domains of different Internet Service Providers—ISPs (X, Y, Z) according to Table 3. Regarding the service representation, an Iperf server was used for different traffic of Constant Bit Rate (CBR).

Table 3 – Technical specification for slices composition. Based on (ETSI, 2020) and (ETSI, 2021).

Slice [1]	**Type**	**E2E Latency (ms)**	**Reliability (%)**	**Data Rate (Mbps)**
1 (VN N1 $\rightarrow$ N4)	Remote Driving	5 (max.)	99.999 (min.)	DL: 1 (min.) UL: 25 (min.)
2 (VN N2 $\rightarrow$ N5)	Rural Macro	Not specific	Higher than 80%	DL: 50 UL: 25
3 (VN N3 $\rightarrow$ N6)	Wireless Road-Side Infrastructure Backhaul (ITS)	30 (max.)	99.999	10

[1] VN: Virtual Network; DL: Downlink; UL: Uplink.

Source: elaborated by the author.

As a result, for each 11 collections performed in the simulation process, a trace file was obtained for each UE containing a dataset of 33 iterations. Iteration data comprises: the amount of transferred bytes, bandwidth used, transport protocol, latency, jitter and packet loss. Each collection was carried out for a period of 5 min. These raw log data are thus the starting point for carrying out the data analysis process.

5.2.2 Test Setup 02

In this test setup, we evaluate another strategy for slice selection. The scenario and methodology are the same as those described in section 5.2.1. However, in this test setup fuzzy logic was employed with the MCDM methods that presented the best performance in Test Setup 01. The aim is to see if there is a significant performance gain in the slice selection method accuracy. The proposed fuzzy system implemented is basically a system that consists of seven inputs and three outputs. The parameters used to define the fuzzy sets, in terms of QoS, are *"Latency", "Jitter", "Loss", "Bandwidth", "Transfer", "Distance", "Reliability"*.

The fuzzy model is based on triangular, trapezoidal or Gaussian membership functions by using the Mamdani model (CASTILLO; CASTRO; MELIN, 2022), (MORDESON; MATHEW, 2018). The parameters for each linguistic variables considered in the membership functions can be visualized in Figures 12 and 13. For the linguistic variable "Reliability", there was no enough precision from the computer used to represent the implementation in this testbed.

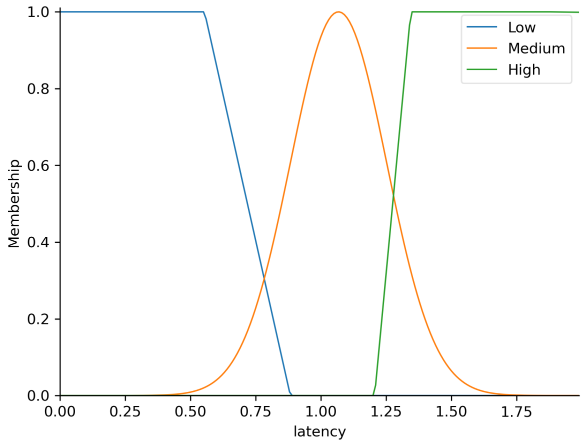
Table 4 presents the intervals found in the dataset, which are considered as a directing factor for linguistic variables construction. However, the construction considers upper and lower bounds on an open interval. In this way, the system accepts and treats values that are not common to the system, the outliers.

Table 4 – TS02: Intervals for each attribute of the original dataset (non-normalized)

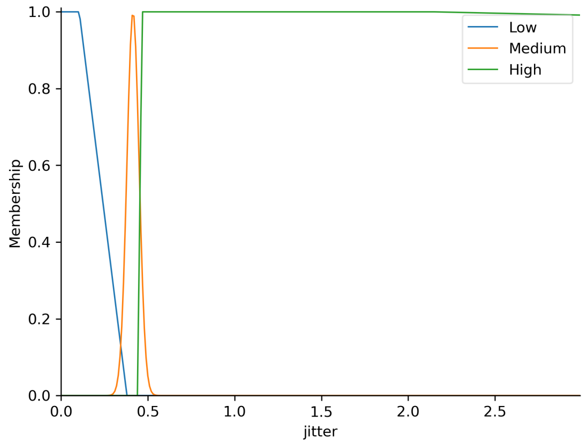
Attribute	**Limit Inferior**	**Limit Superior**	**Average**
Latency	0.554	1.878	1.067772
Jitter	0.104945	2.148467	0.414752
Loss	0	31	4.115286
Bandwidth	32.507692	70000.0	18443.505199
Transfer	139.0	2503000.0	658836.094184
Distance	6.025444	97.929760	53.449390
Reliability	99.984021	99.998972	99.992160

Source: elaborated by the author.

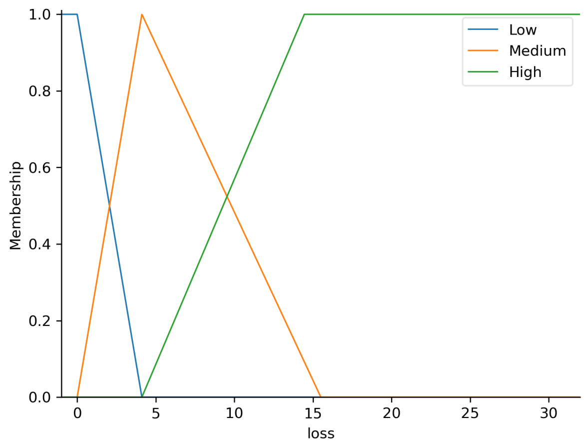
Figure 12 – Fuzzification process: Latency, Jitter and Loss



(a) Latency



(b) Jitter



(c) Loss

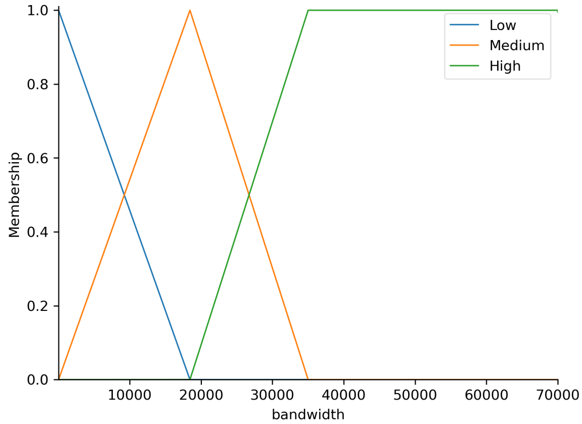
Source: elaborated by the author.

An important factor for fuzzy logic use, about the clustering strategy provided by the K-Means algorithm, as described in section 5.2.1, is the elimination of data normalization process in the processing step, as well as the need for an exclusive use of numeric data, i.e., non-categorical. The fuzzy system is more susceptible to noise when the upper and lower limits for each input attribute are taken into consideration.

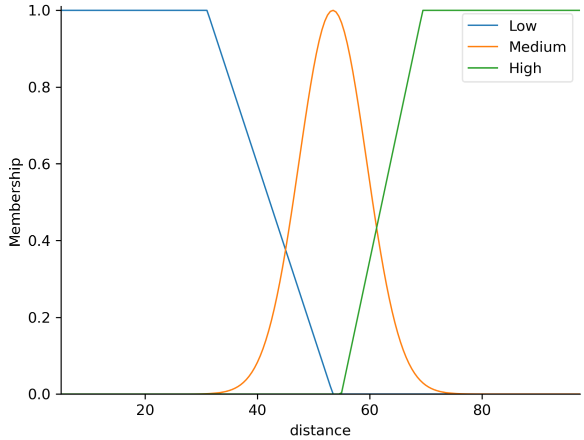
Regarding the defuzzification process, we consider the Center of Maximum (COM) method, where the slices are classified by the parameters obtained by the NSSF DAF' *Collector* module.

The system output is the indicator of the most suitable slice for the user's equipment, having the variable *Slice_ Out* as the output variable. The representation of the output is presented in Figure 14, where it is possible to observe that the system will always recommend a slice to the user. The slices considered are the same in the Table 3 in the

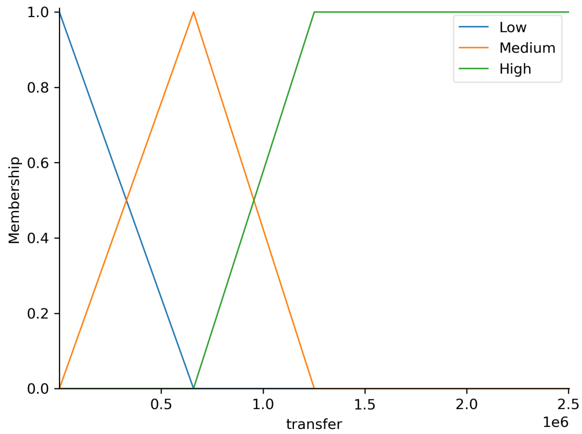
Figure 13 – Fuzzification process: Bandwidth, Distance and Transfer



(a) Bandwidth



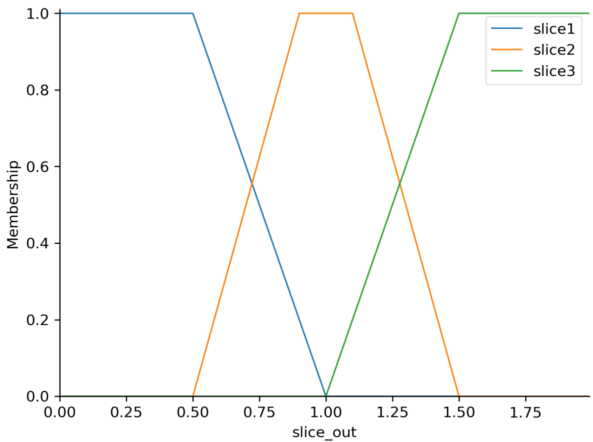
(b) Distance



(c) Transfer

Source: elaborated by the author.

Figure 14 – Defuzzification process: System outputs.



Source: elaborated by the author.

Section 5.2.1.

5.2.3 Test Setup 03

This section presents a hybrid strategy that involves several techniques to implement the NSSF, using the resources and possibilities offered by the MEC, and validating it for IoT service scenarios with wide range of traffic profiles.

Some current services need to migrate to business models based on more cost-effective information delivery and interaction, and the 5G system can help in the adoption of more efficient models to reduce administrative and supply costs (WIJETHILAKA; LIYANAGE, 2021).

Thus, 5G technology can have a particularly important impact on IoT services and applications, for different purposes. In the following, four scenarios of IoT services and applications are described, based on 3GPP Technical Specification TS 22.261 version 18 (3GPP, 2021), which specified characteristics and requirements for dozens of IoT services in the 5G domain.

5.2.3.1 IoT real-time health system

IoT real-time health system involve medical critical applications, i.e., medical devices and applications involved in hospital care, which include vital signal monitoring, cooperation in critical situations, remote surgery procedure using high quality image and augmented reality system, and tele-diagnosis or tele-monitoring systems (WIJETHILAKA; LIYANAGE, 2021; Haghi Kashani et al., 2021). These applications and services, with mission-critical communication characteristics, produce a variety of traffic profiles that must be served by the 5G network.

5.2.3.2 IoT Smart Home System

Smart Home Systems lead people to automate their daily activities. This brings cost reduction and energy preservation, which are major advantages of IoT utilization (HUSSEIN, 2019). Smart home applications range from a simple temperature sensing system that automates the functionality of the air conditioner to a shopping list of supermarket items based on the refrigerator, or an image processing system that ensures home security by identifying potential intruders (MALCHE; MAHESHWARY, 2017). In other words, the diversity of information generated and transmitted can range from a few bits to a high-resolution video stream of monitoring (3GPP, 2021).

5.2.3.3 IoT system for Geohazard prevention (monitoring and early warning)

Geographic hazards include slope deformation, which can be characterized as landslides, debris flow, rockfall, ground surface deformation, surface collapse, surface cracks, and, characterizing a more severe event, earthquake (QIN et al., 2021).

Monitoring risk areas for these events is of utmost importance, preventing life losses from happening by taking preventive evacuation measures. Monitoring systems with low cost displacement sensors can be implemented with the use of WSN. This system generates information data of few bits per second per sensor, but, overall, this adds up to a large volume of data acquisition to be transmitted and analysed, so that it guarantees that the alarms sounds in the eminence of an event that can put people's lives at risk (MEI et al., 2020). This alarm signal should have top priority and be sent to the central monitoring station in a timeframe of a few tens of milliseconds for rapid dispatch of emergency resources.

5.2.3.4 IoT Vehicular System

IoT has tightly coupled with several areas in transportation systems. Four different types of communication modes of V2X are identified by 3GPP: V2V, V2I, V2P and vehicle-to-network (V2N). This variety of vehicular services generates a wide range of traffic profiles that must be served by the 5G network (MEI; WANG; ZHENG, 2019).

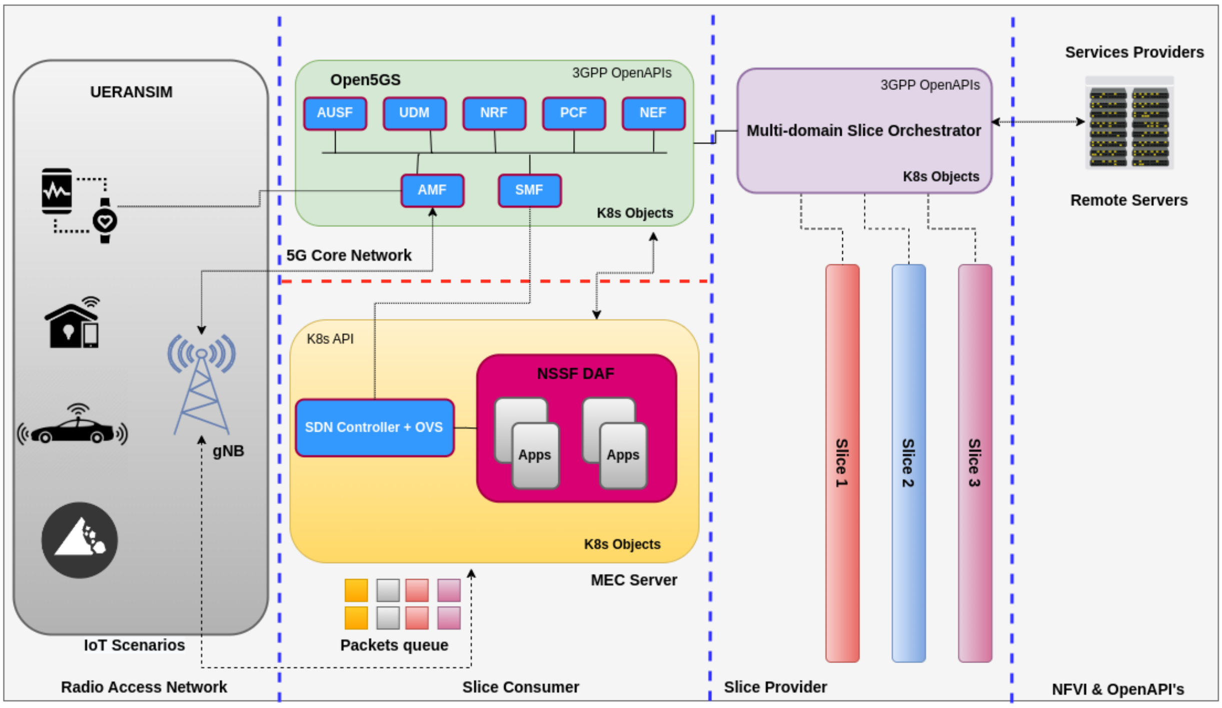
Furthermore, as a critical communication system, it has high reliability and low latency requirements (WIJETHILAKA; LIYANAGE, 2021).

5.2.3.5 Environment description

Figure 15 represents the E2E 5G architecture proposed and evaluated in this work. Note that in the access network there are four IoT scenarios, representing specific groups of services, namely: real-time health system, smart home, V2X and Geohazard prevention. All services have a gateway (broker) that is directly connected to the gNodeB. The UPF receives and identifies the flow tables and assigns the labels according to the service data. Then the packets are forwarded to the edge solution which runs on the MEC Server.

The test environment was implemented by using the K8s Cluster to provide the infrastructure for the virtualized network functions (NFV Infrastructure). In addition, there

Figure 15 – Illustration of the simulation environment architecture for IoT Scenarios with Open5GS (3GPP Release-16) and UERANSIM.



Source: elaborated by the author.

was the implementation in the gNodeB through UERANSIM[15] simulator. A docker container was used with the deployment of the SDN controller OpenDayLight, and Open vSwitch(OvS)[16], to implement the UPF, responsible for data flow separation and identification.

The UERANSIM simulator implements the interface specifications defined in 3GPP TS 38.413[17], which defines the main fields necessary for the identification of a 5G New Radio (NR) cell, in addition to implementing the necessary interfaces for connectivity between the UEs, the edge and the 5G Core Network. For this, it implements the NG Application Protocol (NGAP), Stream Control Transmission Protocol (SCTP) and GPRS Tunneling Protocol User Plane (GTP_U) protocols (FAROOQUI et al., 2022).

The UPF has a public IP, responsible for exchanging traffic with the external network, i.e. outside the simulation environment (UERANSIM + Open5GS). This feature emulates a real traffic exchange with the internet (public network). This physical interface is connected to a Switch 10GB; the other interfaces are virtualized and operate in bridge mode, one of them being connected to the POD that implements the SMF network func-

[15] https://github.com/aligungr/UERANSIM
[16] https://www.openvswitch.org/
[17] https://www.3gpp.org/DynaReport/38413.htm

tion, and another interconnected to the gNB instance inside the UERANSIM simulator, using the standardizes interface N3 and the GTP_U protocol (CHOUDHARI; PATIL; SARAF, 2022).

There is an interface connected to gNB for each IoT Broker. This interface is called "uesimtun0", and implements the GTP protocol for sending packets. The other interfaces are connected to the Calico network provided by the K8s Cluster, according to the specification defined in the implementation model of Section 4.1.3. It is worth emphasizing that the communication between gNB and the AMF network function is carried out through the N2 interface implementation, and uses the NGAP and SCTP protocols (BARRACHINA-MUñOZ; PAYARó; MANGUES-BAFALLUY, 2022).

Note that UERANSIM implements a "slice" field in the messages header coming from UEs. This list is static and predefined according to the slices supported by gNB. In this testbed, this functionality will be implemented by the NSSF DAF - Client, and will occur dynamically through the consumption of APIs with the solution implemented at the edge.

In this test setup, 2000 flows generated four IoT Broker 5G using one thread were considered. For each flow it checks whether the NGAP authentication response messages (*ngap.procedure_ code==46*), i.e., messages describing if the connection between the UE and the AMF function, has already been established. Next, the IP address of the UERANSIM UE (IoT Broker 5G), the SCTP protocol destination port, and the other fields of the NGAP protocol header are considered. The only restriction made in the UPF consisted of a verification and a permission to forward only SCTP data chunk packets. The objective was to avoid the forwarding of SCTP HEARTBEAT packets used in the monitoring and maintenance of SCTP connections, according to RFC 4960 (IETF, 2023), avoiding traffic overhead in the slices. Furthermore, defining the best offloading is the IoT Broker responsibility, not a task of the NSSF DAF Client. It is important to emphasize that the NSSF DAF supports DPI (Deep Packet Inspection) mechanisms through the use of *libpcap* and DPDK[18] (Data Plane Development Kit) tools. These tools allow stress test generation on the 5G Core (Open5GS[19]) and Edge-Core (NSSF and UPF) functions.

The orchestration layer is implemented using ETSI NFV OSM, and has three specialized slices available, as per TS 3GPP 22.261 V.18.5.0 (3GPP, 2021). The service provider includes applications to perform processing and monitoring IoT services data. In this scenario it was implemented using the ELK[20] (Elasticsearch, Logstash and Kibana)

[18]https://www.dpdk.org/
[19]https://open5gs.org/
[20]https://www.elastic.co/

Stack.

The Edge Computing solution is composed of NSSF DAF service. The *Collector* module receives the data streams from the IoT brokers and initializes the data analysis, through the evaluation of QoS and QoE metrics. The dataset is assembled and the machine learning models are initialized. It was instanciate an K8s Cluster, to implement the analysis functions and the MLOps pipeline, using the Python 3.X, Numpy, Pandas and Sklearn libraries, according to the implementation model defined in Section 4.1.3.

The *Processor* module is responsible for the best slice selection, and also for the data forwarding to the orchestration layer through multiple APIs. It implements a set of MCDM methods in Python 3 language (version 3.8), as the Aras, Cocoso, Codas, Copras, Edas, Mabac, Mairca, Marcos, Moora, Ocra, Promethee II, Spotis, Topsis, and Vikor methods.

Tables 5 and 6 present the specification of vertical slices. The NSSF solution proposed in this work aims to cover the IoT scenarios defined in the RAN, using the vertical slices formerly instantiated by the orchestrator, as shown in Figure 15.

Based on the context specified in Tables 5 and 6, the assignments with priority data weights by attributes were considered to establish two scenarios: fair and priority, shown in Table 7.

Table 5 – TS03: Slice specification.

Slices	**Experienced data rate (DL)**	**Experienced data rate (UL)**	**Area traffic capacity (DL)**	...
Urban macro	50 Mbit/s	25 Mbit/s	$100 Gbit/s/km^2 (note2)$	...
Broadband access in a crowd	25 Mbit/s	50 Mbit/s	$[3.75] Tbit/s/km^2$	...
Airplanes connectivity	15 Mbit/s	7,5 Mbit/s	1,2 Gbit/s/plane	...

Legend: NOTE 1: For users in vehicles, the UE can be connected to the network directly, or via an on-board moving base station. NOTE 2: These values are derived based on overall user density. Detailed information can be found in (NGMN, 2016).

Source: adapted from TS 3GPP 22.261 V.18.5.0.

Table 6 – TS03: Slice specification - Continued.

Slices	...	**Area traffic capacity (UL)**	**Overall user density**	**Coverage**
Urban macro	...	$50 Gbit/s/km^2 (note2)$	$10000/km^2$	Full network (note 1)
Broadband access in a crowd	...	$[7.5] Tbit/s/km^2$	$[500.000] km^2$	Confined area
Airplanes connectivity	...	600 Mbit/s/plane	400/plane	(note 1)

Legend: NOTE 1: For users in vehicles, the UE can be connected to the network directly, or via an on-board moving base station. NOTE 2: These values are derived based on overall user density. Detailed information can be found in (NGMN, 2016).

Source: adapted from TS 3GPP 22.261 V.18.5.0.

Table 7 – TS03 - Specification of weights per attribute for tests.

Experiment	Latency	Jitter	Loss	Download	Upload	Distance	Reliability	Density
01	0.125000	0.125000	0.125000	0.125000	0.125000	0.125000	0.125000	0.125000
02	0.044356	0.025868	0.336857	0.135427	0.041821	0.286351	0.074844	0.054476

Source: elaborated by the author.

5.3 Data analytics

In the traditional data analysis process, falling in love with the dataset is a part of the first step. Hence, to know and to prepare the data for the mining process, we conducted the tasks of data pre-processing as follows:

1. Simulation data collection;
2. Data selection;
3. Data purification;
4. Dataset construction.

Algorithm 2 details the process discussed above. The experiment considered six attributes organized in the datasets. However, the network provider can add metrics to suit different scenarios as long as it is able to collect data from the network. To validate this principle, two other criteria were added to the dataset, namely reliability and distance. These, in turn, were randomly generated to follow a normal distribution, respecting a pre-defined interval within what normally exists in real scenarios according to the 3GPP TS 22.186 V16.2.0 and 3GPP TS 22.261 V16.14.0 specifications (ETSI, 2020),(ETSI, 2021). The default dataset then has the following attributes:

1. Latency: End-to-end delay (ms);
2. Jitter: Delay variation (ms);
3. Loss: Packets loss;
4. Bandwidth: Number of bits per second (Mbps);
5. Transfer: Amount of data transferred (sent);
6. UE: User Equipment, not used in the model building process;

7. Experiment: experiment id, not used in the mining process;
8. Distance: Shortest path between UE and Edge;
9. Reliability: Capacity network is functional without interruption.

In the pre-processing step, the data are analyzed regarding the type and interval; treatments such as normalization are commonly applied to the records in this phase. The occurrence of treatment and normalization actions applied to the records is common. The data can thus come to serve and be adaptable to the input pattern of the data mining algorithm used in future steps.

The pre-processing phase is performed on all testbeds.

5.3.1 Analysis of Test Setup 01

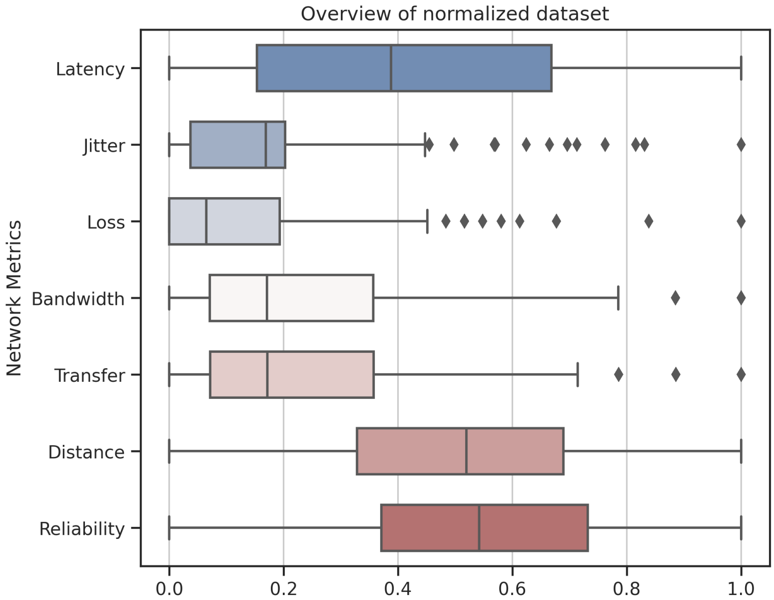
In order to recognize the data, for illustration purposes, Figure 16 presents the network metrics. Note that data were normalized with the use of the *MinMaxScaler* method from the Sklearn library (PEDREGOSA et al., 2011)

The mining process used the K-Means classification or grouping algorithm that uses numerical data. The "protocol" metric is not used because of the distance-based K-Means characteristics. The learning process now considers only the attributes relevant to the experiment: *“Latency”, “Jitter”, “Loss”, “Bandwidth”, “Transfer”, “Distance”, “Reliability”*.

Clustering performs data analysis to recognize the data group behavior so that an element is characterized in one group and differs from the other groups. In the K-Means algorithm, k indicates how many groups are separable in the dataset. One technique to find the correct value of k for a given set is to apply the Elbow Plot Method by calculating the sum of the Root Mean Square Error (SSE) (GANKIDI et al., 2022). The Elbow point occurs when the SSE starts to decrease linearly. Thus, in the dataset used, the appropriate k is 3, as shown in Figure 17, with the SSE for k from 1 to 8.

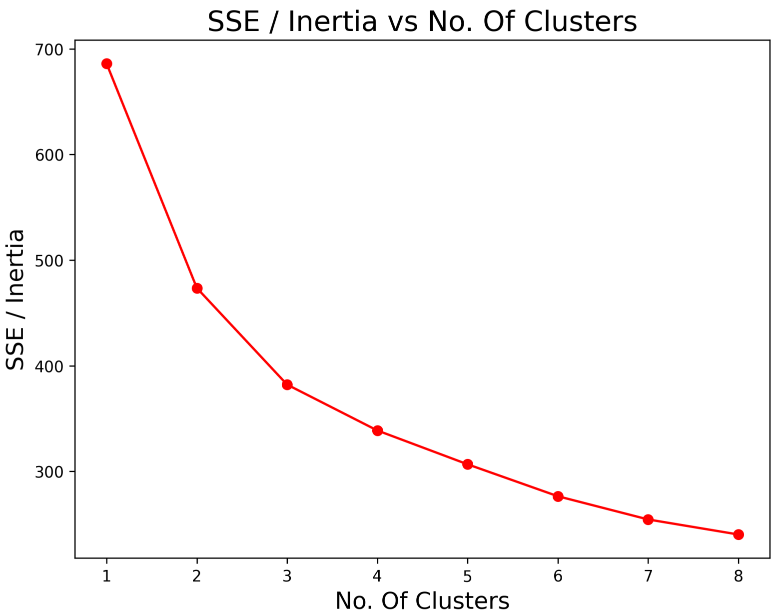
Finally, assuming a k equal to 3, the learning and training process occurs with the separation of elements into three groups that characterize each slice. The next step to data mining is analysis, which seeks to understand the results. To clarify this step, the illustration in Figure 18 presents the unique characteristics of each cluster that, in a way, reflect the characteristics of each slice used in the GNS3 simulation environment. Once the model is trained, the system can receive any data stream within the average training interval to provide the suitable *slice* prediction for the allocation.

Figure 16 – TS01 - Overview of normalized dataset.



Source: elaborated by the author.

Figure 17 – Downward Mean Square Error for K = 1 to K = 8.

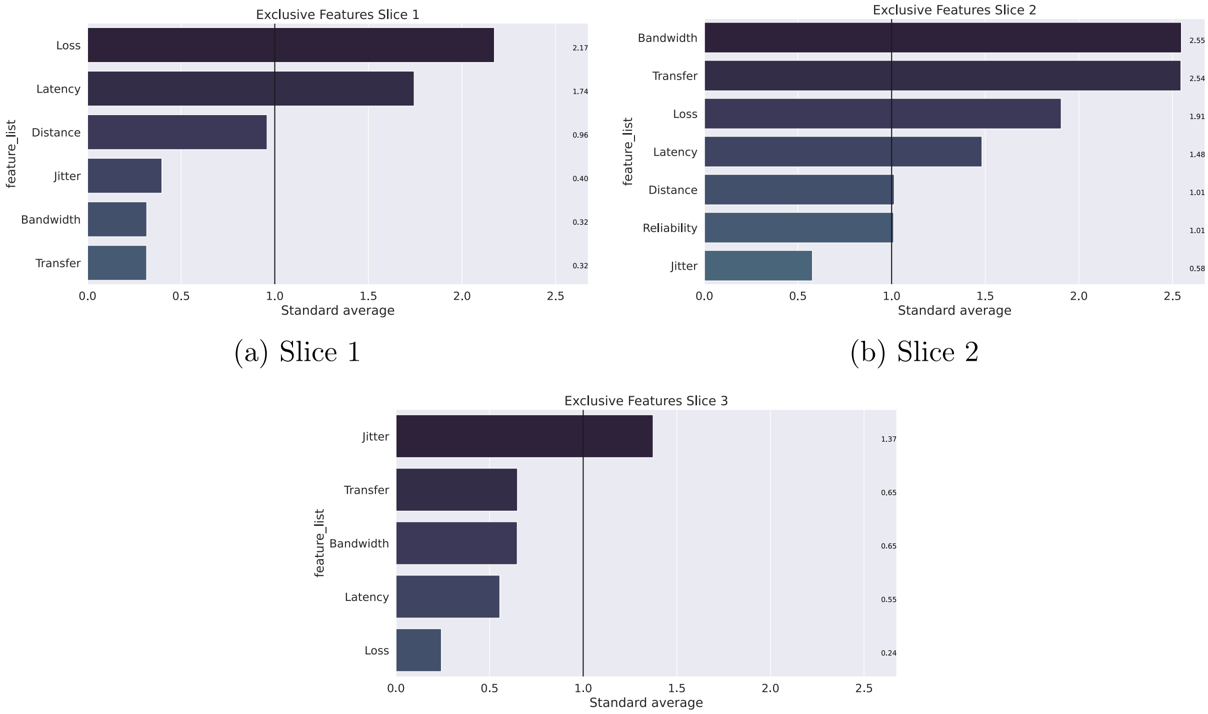


Source: elaborated by the author.

5.3.1.1 Definition of algorithm parameters

After performing the collection and analysis of the network parameters, the application runs for the slice selection. This flow can be seen in the architecture of the proposed framework, illustrated in Figure 4.

Figure 18 – TS01 - Unique characterization of the groups found in the process.



(a) Slice 1

(b) Slice 2

(c) Slice 3

Source: elaborated by the author.

In summary, the configuration of the parameters defined in Algorithm 1 for carrying out the tests is presented in Table 8. The weights were defined in the following order of array parameters *["Latency", "Jitter", "Loss", "Bandwidth", "Transfer", "Distance", "Reliability"]*. The weights configuration followed the results from the K-means algorithm, which showed the most relevant features for each slice (obtained clusters), as illustrated in Figure 18.

Table 8 – TS01 - Weights Setup.

Experiment	**Weights**
1	[0.3, 0.2, 0.1, 0.09, 0.03, 0.03, 0.25]
2	[0.3, 0.4, 0.15, 0.05, 0.05, 0.02, 0.03]
3	[0.10, 0.05, 0.15, 0.3, 0.3, 0.08, 0.02]
4	[0.136, 0.144, 0.144, 0.144, 0.144, 0.144, 0.144]

Source: elaborated by the author.

Thus, for Test 1, the Latency and Reliability criteria were considered with higher weights. Test 2 considered the Latency and Loss criteria, and Test 3 considered the Bandwidth and Transfer criteria. Finally, Test 4 used a fair distribution of weights between the criteria evaluated. Each test is formed by 33 iterations and uses a Flow Table with 250 entries (Flow count).

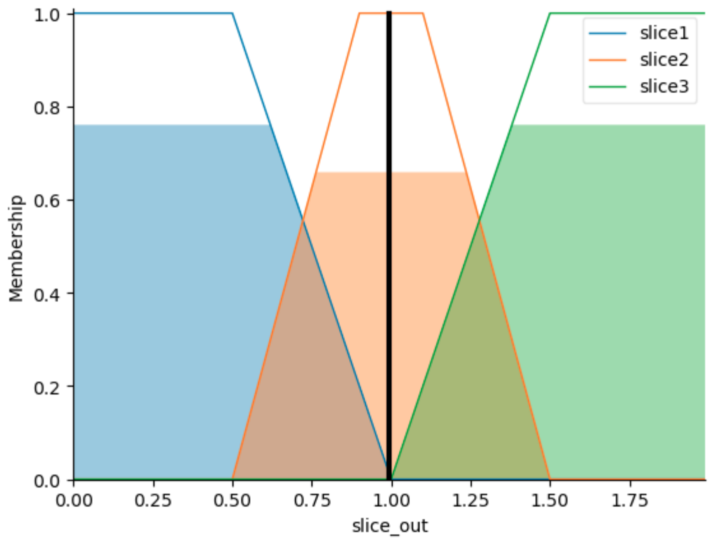
5.3.2 Analysis of Test Setup 02

In this section, the analysis process basically consists of building the inference rules, since the exploratory analysis of the dataset is the same considered in Test Setup 01, as described in section 5.3.1. It is important to note that this approach did not use a vector of weights, since it employs only the relations between the input and output values in the selecting process, according to the rules defined in the Mamdani inference mechanism.

The rules are built from the data grouping analysis in order to map each of the existing slices in the system, as illustrated in Figure 18. For demonstration purposes, four rules are built to map each three existing slices in the system, as described below. However, a total of seventy-five rules were created. For example, in R4 the attributes loss and latency are found in high value in Figure 5, therefore, it is possible to easily perceive that this is an appropriate situation for Slice 1, since this situation is not present for Slice 2 and Slice 3, as can be seen in Figures 18b and 18c.

- Sample of ruleset to select traffic for Slice 1;
  - R1: **IF** loss is low **AND** latency is high **THEN** output is Slice 1
  - R2: **IF** bandwidth is low **AND** transfer is low **AND** jitter is low **THEN** output Slice 1
  - R3: **IF**distance is medium **AND** jitter is low **THEN** output is Slice 1
  - R4: **IF** loss is high **AND** latency is high **THEN** ouput is Slice 1
- Sample of ruleset to select traffic for Slice 2;
  - R5: **IF** bandwidth is high **AND** transfer is high **THEN** output is Slice 2
  - R6: **IF** jitter is low **AND** transfer is high **THEN** output is Slice 2
  - R7: **IF** jitter is low **AND** bandwidth is high **THEN** output is Slice 2
  - R8: **IF** latency is medium **AND** jitter is low **THEN** output is Slice 2
- Sample of ruleset to select traffic for Slice 3;
  - R9: **IF** loss is low **AND** latency is low **THEN** output is Slice 3
  - R10: **IF** bandwidth is low **AND** loss is low **THEN** output is Slice 3

Figure 19 – 5G Slices Selection with the fuzzy system.



Source: elaborated by the author.

- R11: **IF** bandwidth is low **AND**transfer is high **THEN** output is Slice 3
- R12: **IF** jitter is medium **AND** loss is low **THEN** output is Slice 3

An example of the inference process using a specific rule is shown in Figure 19. In this example, the slice selected as the best was Slice 2.

Tests were conducted using genetic algorithms as specified in the *Processor* module, discussed in Section 4.1.1, for automatic generation of inference rules. However, the convergence time made impracticable the application in a system with dynamic characteristics. Thus, there was the decision to resort to a specialist to set up and define the fuzzy rules.

5.3.3 Analysis of Test Setup 03

The data analysis uses data obtained from the testbed illustrated in the Figure 15 and respects the 3GPP specification (3GPP, 2021). For processing purposes, a dataset is constructed with the following attributes: 'Latency', 'Jitter', 'Loss', 'Download', 'Upload', 'Distance', 'Reliability', 'Density'. The flow tables with 2000 flow count for each established scenario are presented in the Tables 5 and 6. In addition, a dataset sample used is shown in Table 9.

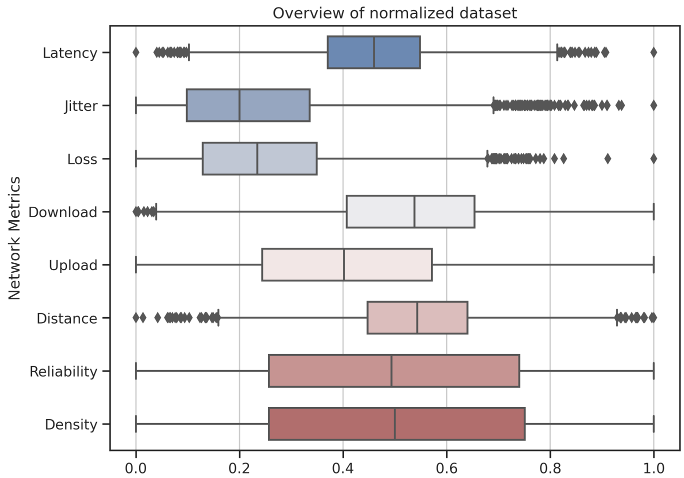
In order to recognize the data, for illustration purposes, Figure 20 presents the network metrics. Note that the data were normalized with the use of the *MinMaxScaler* method

Table 9 – TS03 - Dataset sample.

Jitter	Loss	Distance	Latency	Download	Upload	Reliability	Density
0.360726	1.622586	50.902876	52.781155	44.784232	29.402967	99.999005	3615
0.560079	0.005654	50.671546	52.558219	83.385061	14.347776	99.999309	8501
0.890005	1.838213	58.583555	47.446445	62.431956	23.810858	99.999238	15886

Source: elaborated by the author.

Figure 20 – TS03 - Overview of normalized dataset.



Source: elaborated by the author.

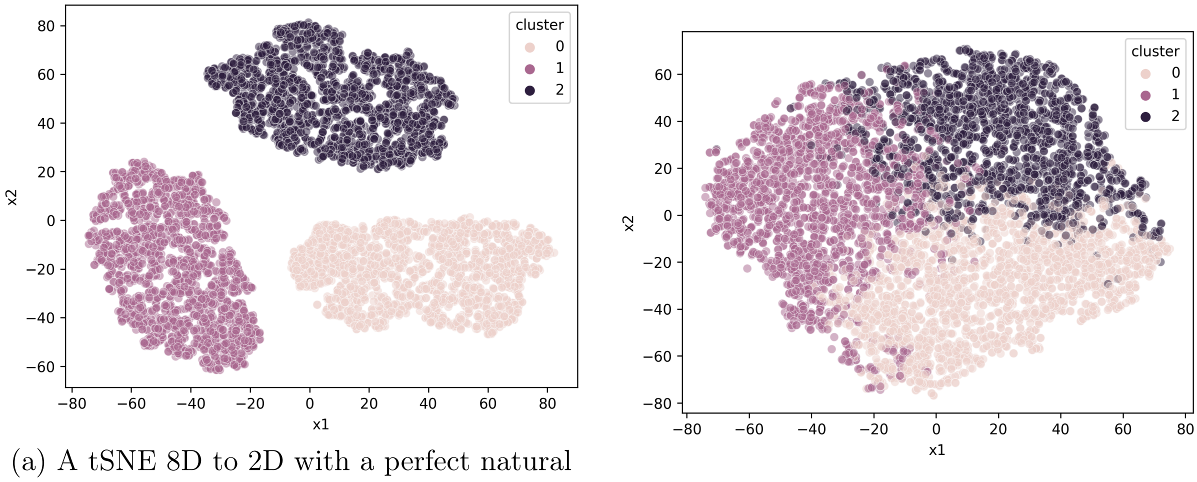
from the Sklearn library (PEDREGOSA et al., 2011).

Initially, the result obtained presented three perfectly defined clusters, as shown in Figure 21a, that use the t-distributed Stochastic Neighbor Embedding (t-SNE) tecnique (LIU et al., 2020). The t-SNE is on the state-of-the-art in computer visualization for high-dimensional data. It is a technique for data dimensionality reduction and extracting local clustering to represent 2D or 3D graphic (PEZZOTTI et al., 2017).

For this reason, the perfectly natural clusters, to avoid the use of an overfitted dataset, we used noise insertion and changing scenario thresholds, allowing the existence of intersection in the range of some attributes. For example, the intervals of three scenarios for the download transmission rate attribute are: min - 50 to 500 Mbps; min - 25 to 220 Mbps and min - 15 to 100 Mbps, respectively. As a consequence, the evaluation of k from K-Means by sum of squared error (SSE) presents two possible candidates, allowing the arbitrary decision for the k-value, as in Figure 22. The flat view of the cluster for k equal to three with adjusted dataset is shown in the Figure 21b.

Finally, by assuming that k equals 3, the trained model is persisted and becomes capable of processing any data stream within the average training interval to provide the

Figure 21 – A tSNE 8D to 2D.

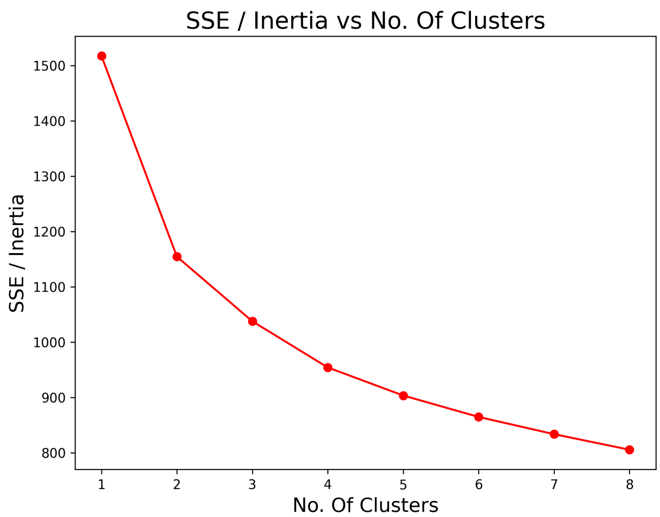


(a) A tSNE 8D to 2D with a perfect natural clustering

(b) A tSNE 8D to 2D with $k = 3$.

Source: elaborated by the author.

Figure 22 – SSE.



Source: elaborated by the author.

prediction of the slice suitable for use. The model's output value has been added as the class attribute to dataset, in a way that allows the use of mathematical models for the decision-making process.

5.4 Chapter summary

In this chapter, three testbeds were implemented in the NSSF DAF validation. The scenarios comprise different technologies, tools, simulation environments, in addition to portraying different slice specifications, defined according to 3GGP, and aiming to meet various services and applications in the 5G context. In this sense, we sought to validate the proposed solution in different scenarios, showing its applicability and its easiness of integration with the main market tools.

The data acquisition, cleaning, and analysis pipeline for each evaluated scenario was detailed, in addition to the adjustments of the algorithms parameters used in each of them. We also sought to represent the different characteristics application in use, the different models of solution deploying, as well as the use of different computational platforms.

The experiments were performed in order to validate different framework features. In addition, we sought to use different virtualization and cloud computing solutions, allowing for the solution validation in environments as close as possible to slices in production.

6 RESULTS

This chapter contains the main results of this research and is divided into the following sections: 6.1 defines the hypotheses and the main statistical tests used in the experiments; 6.2 explains the strategy used in the analysis of the results for the TS-01, in addition to describing and comparing the results of each experiments in this scenario; 6.3 describes and evaluates the gains performed by using conjugated fuzzy logic with the MCDM methods for the TS-02 scenario; 6.4 describes and analyzes the results for the TS-03 scenario, evaluating the results for priority network traffic; and 6.5 concludes this chapter.

6.1 Definitions of Null and Alternative Hypotheses

After performing the experiments specified in the previous chapter, and once the data results from each method could be obtained, a descriptive analysis verified whether there were significant differences in the methods performance for the slices evaluated in the set of tests. The experiment incorporates a comparative analysis, utilizing multiple comparison Tukey's test derived from the analysis of variance. We also applied the Shapiro–Wilk normality, Durbin–Watson independence, and Fligner–Killeen homoscedasticity tests (JAMES et al., 2021), (MONTGOMERY; RUNGER, 2018).

For all tests performed, a significance level $\alpha = 0.05$ or 95% confidence level was considered.

Below, the hypotheses for each statistical tests performed are presented. The equation (6.1) presents the main hypothesis evaluated in this work. The other equations follow the order: i) (6.2) Shapiro–Wilk test; ii) (6.3) Durbin–Watson test; iii) (6.4) Fligner–Killeen test; iv) (6.5) Tukey's test.

For all equations, H_0 corresponds the null hypothesis, and H_a the alternative hypothesis.

$$H_0 : \tau_i = \tau_j \quad \text{(There is no difference between the accuracy of methods in the slice selection process.)}$$
$$H_a : \tau_i \neq \tau_j \quad \text{(There is difference between the accuracy of methods into the slice selection process.)} \tag{6.1}$$

$$H_0 : \theta_i = \theta_j \quad \text{(The sample came from a normally distributed population.)}$$
$$H_a : \theta_i \neq \theta_j \quad \text{(The sample does not come from a normally distributed population.)} \tag{6.2}$$

$$H_0 : \phi_i = \phi_j \quad \text{(The sample has independent residuals.)}$$
$$H_a : \phi_i \neq \phi_j \quad \text{(The sample hasn't independent residues.)} \tag{6.3}$$

$$H_0 : \sigma_i = \sigma_j \quad \text{(The sample presents homoscedasticity of variances.)}$$
$$H_a : \sigma i \neq \sigma j \quad \text{(The sample does not present homoscedasticity of variances.)} \tag{6.4}$$

$$H_0 : \bar{Y}_{B_i} = \bar{Y}_{A_j} \quad \text{(The accuracy of methods "A" and "B" do not differ significantly.)}$$
$$H_a : \bar{Y}_{B_i} \neq \bar{Y}_{A_j} \quad \text{(The accuracy of methods "A" and "B" differ significantly.)} \tag{6.5}$$

6.2 Results for Test Setup 01

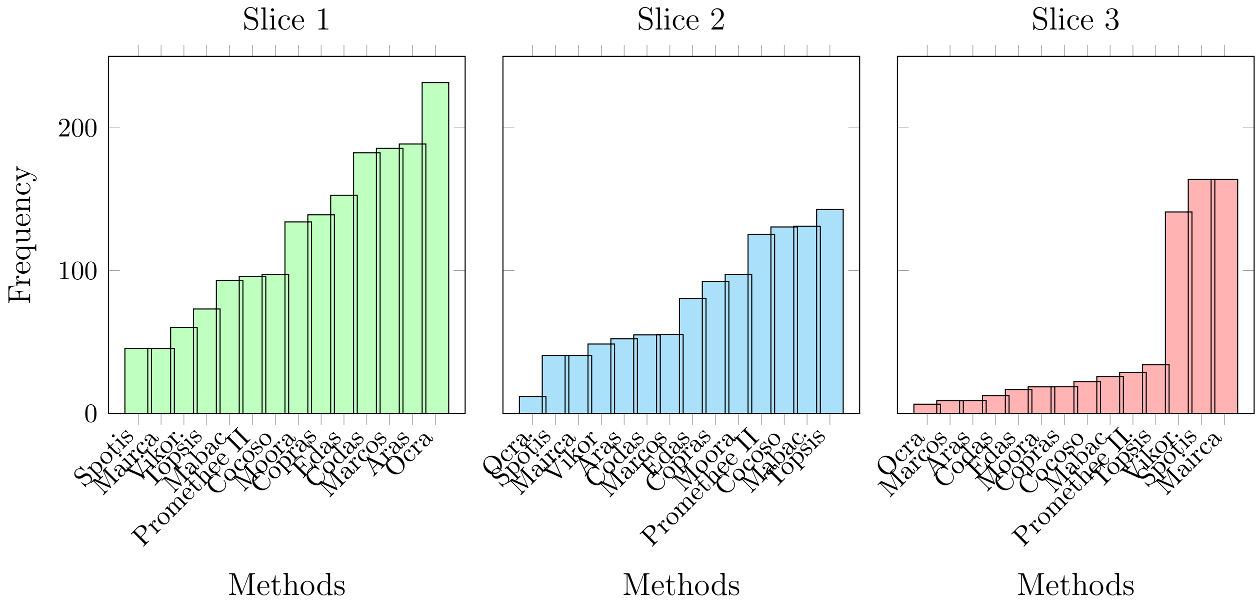
To understand the results measured by each approach, the behavior of the criteria in the network flows was observed. To this end, we used the boxplot statistical tool to evaluate the main QoS criteria, considering the data symmetry, its dispersion, and the possible presence of outliers. Additionally, we used the bar graphs feature and network graphs to verify the performance of the methods considered, finding to validate the hypothesis defined in the equation (6.1).

Note that Figure 16 presents several outliers for the jitter and loss criteria. This is characterized by their stochastic, dynamic, and nonlinear behavior and the variation in traffic rates considered in Table 2.

After this preliminary behavior analysis of the criteria adopted as input parameters, we evaluated the fourteen approaches concerning their ability of classifying the best slice, considering the weights or preferences defined a priori by the *Decision Maker*, and also the UEs traffic requirements.

For a better data visualization in the tables, rounding to two decimal places was

Figure 23 – Behavior of the NSSF DAF methods - TS 1: Experiment 1



Source: elaborated by the author.

considered. It should be noted that the TS 3GPP 22.261 v.16.14.0 specification defines different characteristics for the specialized configuration of each slices (ETSI, 2021). In this work, as described in Table 3 in Section 5.2.1, Slice 1 is specialized in serving traffic referring to 'Remote Driving', Slice 2 in 'Rural Macro', and Slice 3 in ITS traffic.

6.2.1 Experiment 1

Table 10 details the results for each method used in the NSSF DAF service to the Experiment 1. This table results for mean values are summarized in Figure 23.

In Experiment 1, the following attributes were prioritized: *"Latency", "Jitter", "Reliability"*, according to the weights defined in Table 8. The weights correspond to the result of the K-means algorithm analysis, discussed in Section 5.3.1. This configuration prioritizes Slice 1. In this sense, we seek to verify the MCDM methods sensitivity to the slice requirements of the 'Remote Driving' type, that is, the methods that demonstrate higher accuracy in selecting Slice 1 will be considered superior in this experiment.

It was observed, according to Table 10 and Figure 24a, that among the evaluated methods, the accuracy of the Ocra method stands out, which showed greater consistency in the selection of Slice 1, presenting a low variance and standard deviation, in addition to having a more homogeneous distribution for the 33 iterations considered in the experiment. The Aras, Codas and Marcos methods also demonstrated high averages, however, with a

Table 10 – Methods Results: TS01 - Experiment 1

Slice	Methods	Mean	STD	Var	CI
Slice 1	Aras	188.70	19.38	375.47	181.83 –195.57
	Cocoso	97.18	28.83	831.03	86.96 – 107.40
	Codas	182.52	21.62	467.32	174.85 – 190.18
	Copras	139.09	45.90	2106.84	122.82 – 155.37
	Edas	152.79	43.56	1897.23	137.34 – 168.23
	Mabac	93.00	43.82	1920.31	77.46 – 108.54
	Mairca	45.58	24.61	605.63	36.85 – 54.30
	Marcos	185.61	21.78	474.37	177.88 – 193.33
	Moora	134.12	47.64	2269.23	117.23 – 151.01
	Ocra	231.64	3.90	15.18	230.26 – 233.02
	Promethee II	95.91	38.62	1491.15	82.22 – 109.60
	Spotis	45.58	24.61	605.63	36.85 – 54.30
	Topsis	73.15	48.63	2364.76	55.91 – 90.39
	Vikor	60.30	24.06	579.03	51.77 – 68.84
Slice 2	Aras	52.24	25.61	655.75	43.16 – 61.32
	Cocoso	130.58	32.90	1082.31	118.91 – 142.24
	Codas	55.00	30.05	902.75	44.35 – 65.65
	Copras	92.24	51.76	2679.38	73.89 – 110.60
	Edas	80.45	45.19	2041.88	64.43 – 96.48
	Mabac	131.12	58.47	3418.17	110.39 – 151.85
	Mairca	40.64	41.63	1732.74	25.88 – 55.40
	Marcos	55.39	27.28	744.43	45.71 – 65.07
	Moora	97.27	52.98	2806.77	77.49 – 116.06
	Ocra	11.91	4.80	23.09	10.21 – 13.61
	Promethee II	125.30	50.22	2521.72	107.50 – 143.11
	Spotis	40.64	41.63	1732.74	25.88 – 55.40
	Topsis	142.79	75.44	5691.80	116.04 – 169.54
	Vikor	48.61	56.21	3159.56	28.67 – 68.54
Slice 3	Aras	9.06	11.10	123.25	5.12 – 13.00
	Cocoso	22.24	8.40	70.56	19.26 – 25.22
	Codas	12.48	14.02	196.70	7.51 – 17.46
	Copras	18.67	17.15	294.23	12.58 – 24.75
	Edas	16.76	14.80	219.06	11.51 – 22.01
	Mabac	25.88	28.07	787.73	15.93 – 35.83
	Mairca	163.79	49.38	2438.23	146.28 – 181.30
	Marcos	9.00	10.18	103.69	5.39 – 12.61
	Moora	18.61	18.18	330.37	12.16 – 25.06
	Ocra	6.45	2.53	6.38	5.56 – 7.35
	Promethee II	28.79	21.63	468.05	21.12 – 36.46
	Spotis	163.79	49.38	2438.23	146.28 – 181.30
	Topsis	34.06	39.95	1595.75	19.90 – 48.23
	Vikor	141.09	59.39	3526.77	120.03 – 162.15

Legend: STD: Standard Deviation; Var: Variance ; CI: Confidence Intervals.

Source: elaborated by the author.

greater variance, and showing a greater number of outliers.

The descriptive analysis checked for significant differences in the methods performance for the slices evaluated in the tests set, considering a confidence level of 95%. In this sense, the parametric methods defined in the equations of Section 6.1 were used. The results from the Shapiro–Wilk normalization test show that the data set does not follow a normal distribution. Using the Durbin-Watson test, the sample variables do not present independent residues. From the Fligner-Killeen test, the sample does not present homoscedasticity of variances. Thus, the null hypothesis was rejected for all tests considered.

Due to the violation of normality, and the great variability of the response variable (Frequency of Selection), as illustrated in Figure 24, it was impossible to compare the methods using an ANOVA model, and therefore, the test application of Tukey HSD or Bonferroni multiple comparisons, as the p-value response in these tests reaches the maximum difference and is fixed at "1".

Therefore, it was decided to apply the KRUSKAL-WALLIS non-parametric test (JAMES et al., 2021). However, parametric tests are less efficient when compared to parametric methods, they help to give a broader view of the comparison between the evaluated MCDM methods. From Table 10, considering Slice 1, it was verified that the median of the evaluated methods are different, and the Ocra method presents the lowest interquartile range (IQR), followed by the Aras, Codas and Marcos methods, which corroborates with the descriptive analysis illustrated in Figure 24a.

The stochastic nature of the problem, in addition to the lack of guaranteed "predictability", makes it complex to create models with a better fit to the data. Furthermore, the considered response variable has characteristics of counting data, which suggests the use of a binomial or negative binomial distribution (JAMES et al., 2021). It was verified that it would not be feasible to use a negative binomial distribution, due to the fixed number of samples, 250 network flows were considered, i.e., there is no variation in this number. We then proceeded to evaluate the Bernoulli distribution (MONTGOMERY; RUNGER, 2018), using the simulated envelope graph, as illustrated in Figures 26–28. Residual plots are a useful graphical tool for identifying non-linearity and can be used to identify outliers too.

From the analysis of Slice 1, note that, although the response variable has count characteristics, it was not possible to obtain a data fit using a logistic regression model (GLM). Thus, a linear regression model (LM) was used, applying the $sqrt()$ function to the response variable. The application of the $sqrt()$ function makes the results more reasonable, i.e., the number of successes approached a normal distribution, as illustrated in Figure 26b. This procedure allows summarizing the result of the linear regression model in an ANOVA, and therefore applying the TukeyHSD test, described in the equation (6.5).

From Tukey's test, it was observed that there is no significant difference between the various methods. The comparison is performed in pairs between all methods (182 combinations). Given the large number of comparisons, the visualization from Tukey's graph became impracticable. To solve this problem, a network graph was used, as shown in Figure 25a, where two clusters can be seen, which corresponds to the methods that do not have significant differences in performance, considering a confidence level of 95%. The network graph supports the descriptive analysis data defined in Table 10 and Figure 24a. In the other comparisons, there are significant differences in the accuracy of MCDM methods.

With regards to the analysis estimated coefficients in the regression model, for Slice 1 it is observed that the Ocra method reached the highest positive number (1.50), which demonstrates its superiority comparing to the others. The confidence interval for the Ocra method is between 0.65 and 2.35.

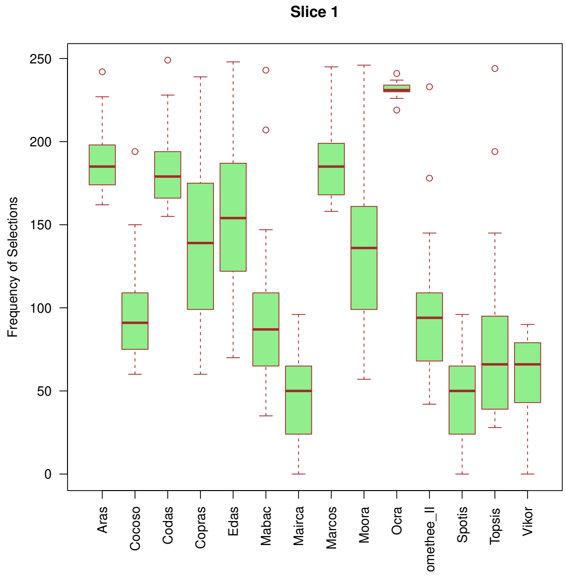
For Slice 2, the same analysis strategy as for Slice 1 was used. The null hypothesis was rejected for all applied tests. Thus, there was a violation of the premise of normality and independence of the residues, and also a violation of the premise of variances homoscedasticity. As a consequence of these violations, the KRUSKAL-WALLIS non-parametric test was used. In this test, it was observed that the median of the evaluated methods are different, i.e., for Slice 2 there are also differences in the performance of the evaluated methods. Additionally, it was found that the Ocra method had the lowest IQR.

To enable multivariate comparison between the methods, a linear regression model was employed, according to the model fit shown in Figure 27. Then, the result was summarized in an ANOVA, allowing the Tukey test application. The network graph illustrated in Figure 25b shows the result of comparing Tukey's method. Note that of 182 comparisons done, in 45 of them, there is no significant difference between the methods.

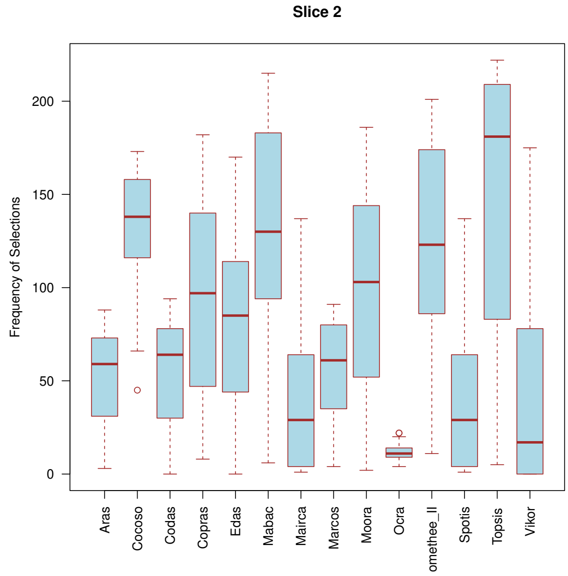
With regards to the analysis of the estimated coefficients in the regression model, it was observed that the Topsis (4.42), Cocoso (4.41), Mabac (4.15) and Promethee II (4.00) methods demonstrated superiority in the selecting accuracy for the Slice 2. This performance can be observed in the descriptive analysis illustrated in Figure 24b.

For Slice 3, there was also rejection of the null hypotheses defined in equations (6.2), (6.3), and (6.4). Based on the approach taken in the analysis of Slices 1 and 2, it was

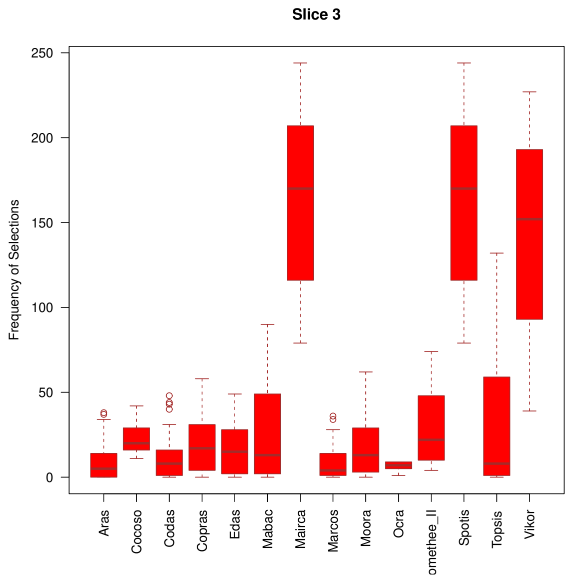
Figure 24 – Descriptive analysis using boxplot - TS 01: Experiment 1.



(a) Slice 1



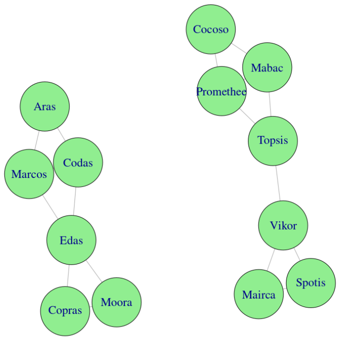
(b) Slice 2



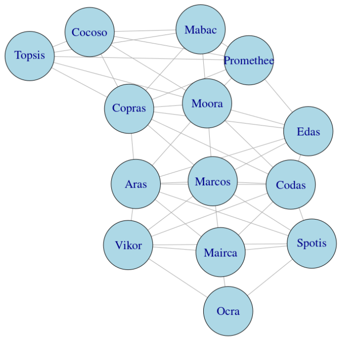
(c) Slice 3

Source: elaborated by the author.

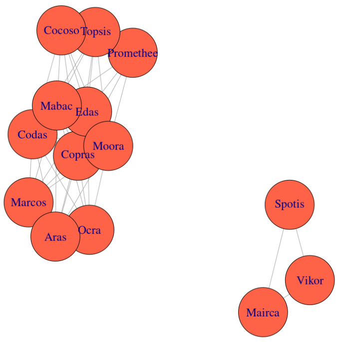
Figure 25 – Network graph TS01: Experiment 1. There is no significant difference between this methods.



(a) Slice 1



(b) Slice 2

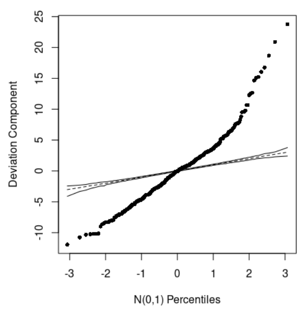


(c) Slice 3

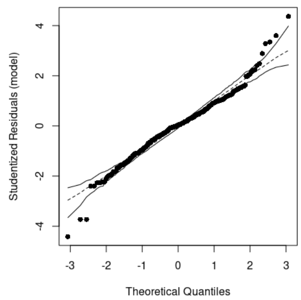
Legend: 95% confidence intervals comparing each pair of methods.

Source: elaborated by the author.

Figure 26 – Simulated envelope TS01: Experiment 1 - Slice 1.



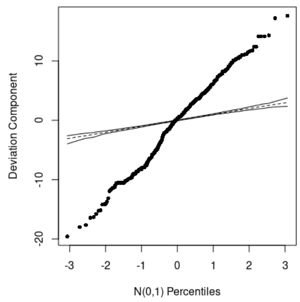
(a) Binomial GLM.



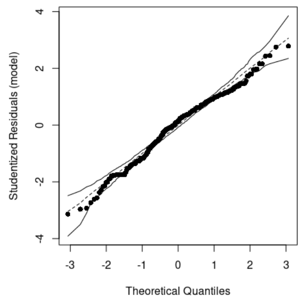
(b) Normal LM.

Source: elaborated by the author.

Figure 27 – Simulated envelope TS01: Experiment 1 - Slice 2.



(a) Binomial GLM.



(b) Normal LM.

Source: elaborated by the author.

observed that for Slice 3 there are also differences in the performance of the evaluated methods. Using the adjustment of the regression model illustrated in the simulated envelope graph in Figure 28, and then applying Tukey's test, it was found that from the 182 comparisons completed, 48 did not have a significant difference in terms of the accuracy of selection of Slice 3. This behavior can be observed in Figure 25c.

In terms of the estimated coefficients analysis in the linear regression model, it is observed that the Spotis (10.32), Mairca (10.32) and Vikor (9.26) methods, demonstrate superiority in the selection accuracy for Slice 3. However, this finding is inconsistent with the other methods, since for experiment 1 the attributes with the highest weight were Latency and Jitter, and considering the slices characteristics from the machine learning model used, Slices 1 and 2 with the highest probability of being selected. These results indicate that the Spotis, Mairca and Vikor methods show greater adherence to traffic destined for Slice 3. These results can be confirmed in Table 10 and Figure 24c.

The findings presented in this section are further complemented by the results shown in Appendix B (experiments 2,3 and 4). The experiments shown in Appendix B differ from those presented in this section by their weight setup.

6.3 Results for Test Setup 02

To conduct the experiments in Test Setup 02, the Ocra, Aras and Codas methods were considered. They obtained the best performance in all the TS-01 experiments for Slice 1, in particular the tests conducted in Experiment 1, as detailed in Section 6.2, and in Subsection 6.2.1. Slice 1 was used as a reference because it has the highest QoS

Figure 28 – Simulated envelope TS01: Experiment 1 - Slice 3.

(a) Binomial GLM. (b) Normal LM.

Source: elaborated by the author.

requirements for addressing flows for Remote Driving scenarios (e.g. V2X and IoV), considered one of the most complex environments for 5G and future networks. These scenarios consider different levels of automation (MOLINARO et al., 2020),(MEI; WANG; ZHENG, 2019),(CONDOLUCI et al., 2019), (CAMPOLO et al., 2018).

The dataset used in this test setup are the same considered in TS-01, as described in Section 5.3.1 of Chapter 5. Thus, the same criteria of Experiment 1 in TS-01 were prioritized, as defined in Table 8. They are Latency, Jitter and Reliability. In this configuration, Slice 1 is prioritized. In addition, a fair distribution of weights among all criteria was also considered, which in TS-01 is carried out in Experiment 4.

In this test setup, we sought to verify whether there were gains in the Ocra, Aras and Codas methods when combined with fuzzy logic in the NSSF DAF, for the treatment and addressing of flow tables from the UPF. Experiment 1 uses the weights defined in Table

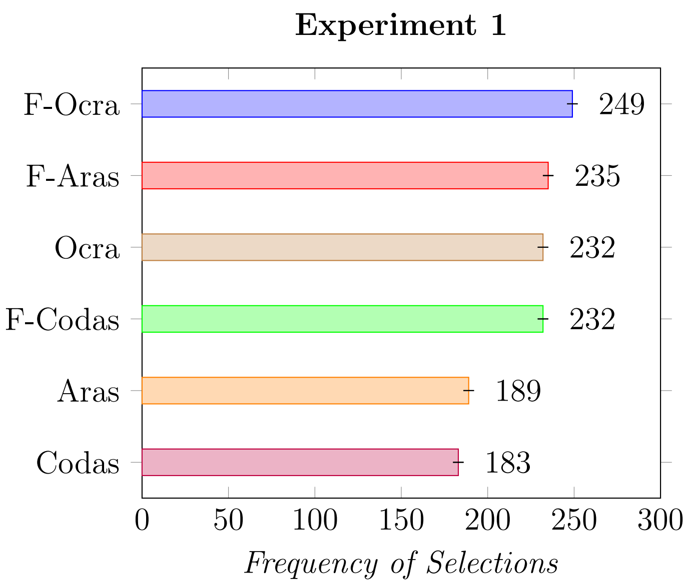
Table 11 – Methods Results: TS02: Performance Gains - Experiments

Experiment	**Methods**	**Mean**	**STD**	**Var**	**CI**
01 - Gains (%)	F-Aras	24.52	7.75	60.07	21.77 - 27.27
	F-Codas	26.99	8.65	74.77	23.93 - 30.06
	F-Ocra	7.35	1.56	24.43	6.79 - 7.90
02 - Gains (%)	F-Aras	9.99	6.26	39.19	7.77 - 12.21
	F-Codas	9.39	7.84	61.40	6.62 - 12.17
	F-Ocra	4.93	0.92	0.85	4.61 - 5.26

Legend: STD: Standard Deviation; Var: Variance ; CI: Confidence Intervals.

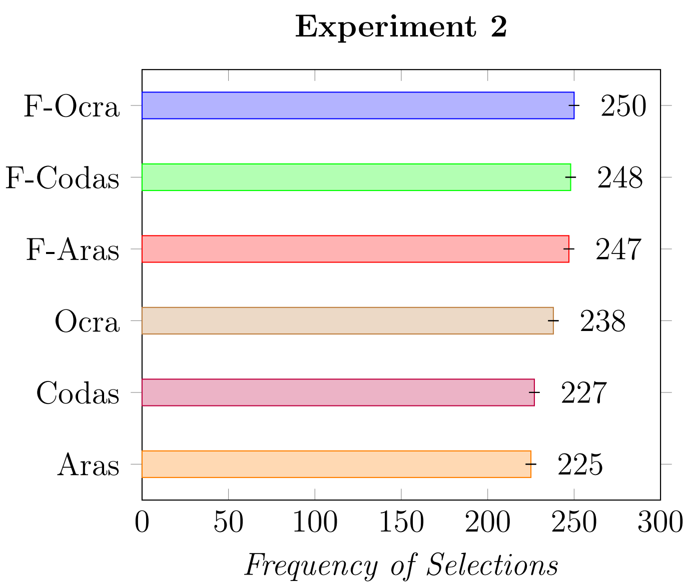
Source: elaborated by the author.

Figure 29 – TS-02: Performance Gains to Experiment 1



Source: elaborated by the author.

Figure 30 – TS - 02: Performance Gains to Experiment 2



Source: elaborated by the author.

8 (see TS-01 Experiment 1), and Experiment 2 assumes a fair distribution of weights.

From the data analysis in Table 11, considering a confidence level of 95% and for a set of 33 iterations and 250 flows tables, it was observed in Experiment 1 that the fuzzy logic provided a performance gain in accuracy of slice selection. The Fuzzy Aras (F-Aras) and Fuzzy Codas (F-Codas) methods obtained a performance gain of more than 24% in selection accuracy, while the Fuzzy Ocras (F-Ocras) method obtained a 7.35% gain. These results are illustrated in Figure 29.

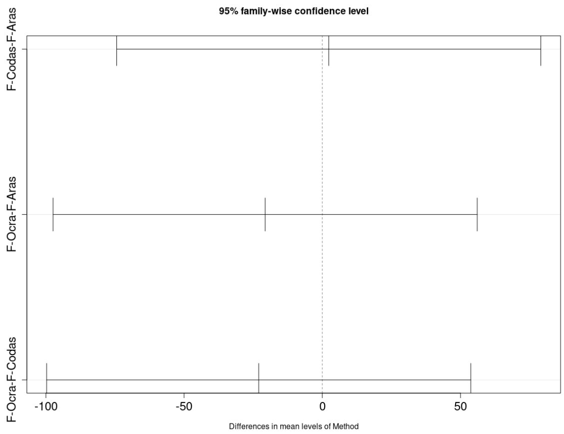
For Experiment 2, the use of fuzzy logic with the MCDM methods promoted an average performance gain of over 9% for the F-Aras and F-Codas methods, while the F-Ocra method had a gain of 4.93%, as detailed in Table 11 and shown in Figure 30.

In addition, we sought to verify whether there were significant differences in the selection accuracy of the F-Aras, F-Codas and F-Ocra methods in both experiments. From Shapiro-Wilk, Durbin-Watson and Fligner-Killen tests, according to the hypotheses defined in equations (6.2), (6.3), and (6.4), it was verified that the tests results accepted the null hypothesis, i.e., for the TS-02 experiments there was no violation neither of the premisses of normality and independence of the residuals, nor of the homoscedasticity of the variances, considering a confidence level of 95%.

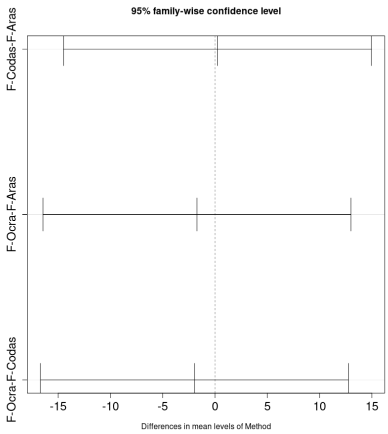
Then, an ANOVA test was applied, considering a significance level $\alpha = 0.05$. The results showed that there is no difference in the mean value of slice selection accuracy between the evaluated methods. Finally, the Tukey's method was applied to perform multiple comparisons. Figure 31 shows the pairwise comparison for Experiment 1, considering the ANOVA model and the adjustment made by the LM model. Note that all confidence intervals overlap with zero, so there are no significant differences considering the defined confidence level. Figure 32 shows the same analysis for Experiment 2. Furthermore, it is observed that there is a flattening of the confidence interval due to the increase in the p-value provided by the LM model.

Finally, as previously highlighted, the use of fuzzy logic associated with the MCDM methods helped to ensure that no violation of the premise of normality was made. Figure 33a shows the analysis of residuals considering the ANOVA model, and Figure 33b shows the adjustment done from the LM model for Experiment 1. Figures 34a and 34b illustrate the analyzes for Experiment 2. Note that outliers still occur, but the model is able to describe adequately the behavior response variable.

Figure 31 – TS 02 - 95% confidence intervals comparing the methods into Experiment 1 with Tukey's Test.



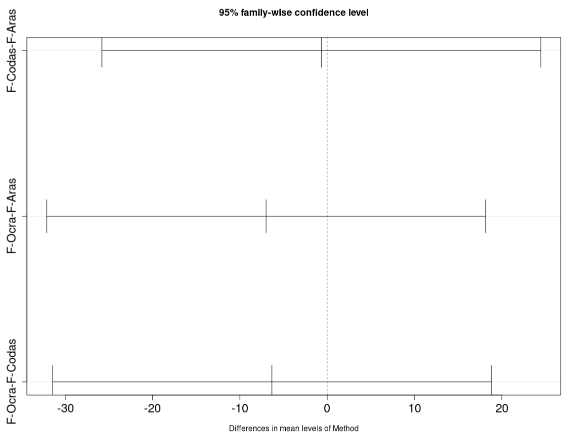
(a) ANOVA Model.



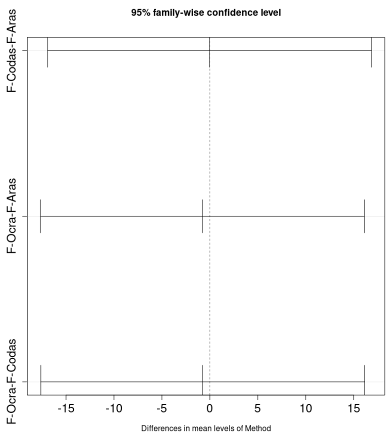
(b) LM Model.

Source: elaborated by the author.

Figure 32 – TS 02 - 95% confidence intervals comparing the methods into Experiment 2 with Tukey's Test.



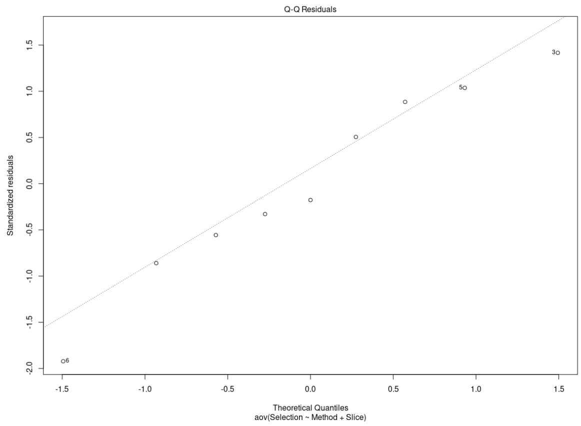
(a) ANOVA Model.



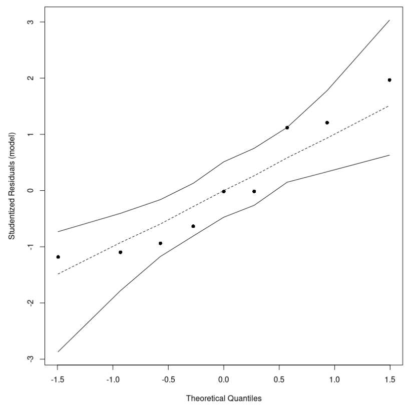
(b) LM Model.

Source: elaborated by the author.

Figure 33 – TS 02 - Residual Plots to Experiment 1.



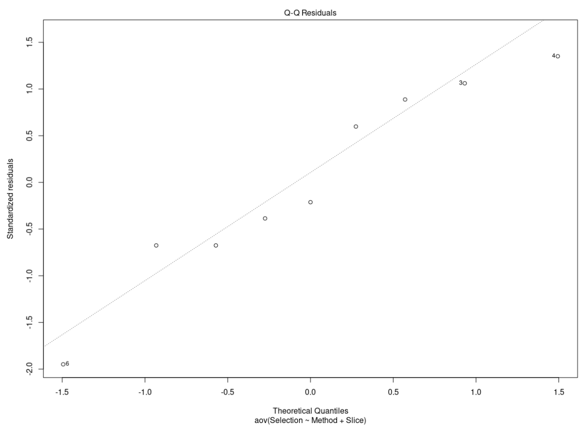
(a) For ANOVA model.



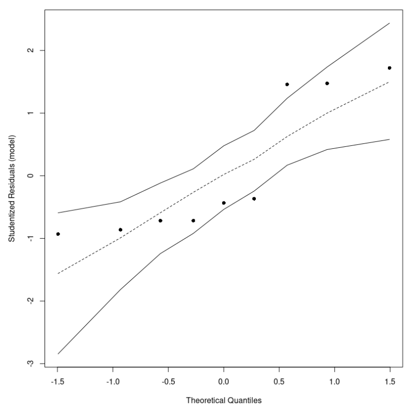
(b) For LM model.

Source: elaborated by the author.

Figure 34 – TS 02 - Residual Plots to Experiment 2.



(a) For ANOVA model.



(b) For LM model.

Source: elaborated by the author.

6.4 Results for Test Setup 03

The purpose of this test setup, as detailed in Section 5.2.3, is to evaluate the NSSF DAF in scenarios involving IoT, i.e., from the use of networking slices for mMTC, following the specifications defined from 3GPP. In addition, we sought to evaluate the proposed solution in scenarios with priority traffic, as it is the case involving patient monitoring and natural disasters. The idea is to validate the hybrid algorithms in an integrated Edge and Cloud Computing architecture for critical application scenarios.

The vertical slices considered were implemented as specified in Tables 5 and 6, and used the scenario in Figure 15. The strategy for analyzing the results was the same used in TS-01, and tested the set of hypotheses defined in Section 6.1. Two experiments are performed. For Experiment 1, a fair distribution of the weights of each criteria (QoS attributes) was considered in the selection of the best slice. For this experiment, the methods could select the best slice according to the traffic conditions in network runtime, and there were no attributes or network metrics being prioritized.

For Experiment 2, the following criteria were prioritized, in this order: Loss, Distance, Download, Reliability, Density, Latency, Upload and Jitter, according to data presented in Table 7. After obtaining the machine learning model, we evaluated the fourteen MCDM methods adopted in the solution in the two experiments.

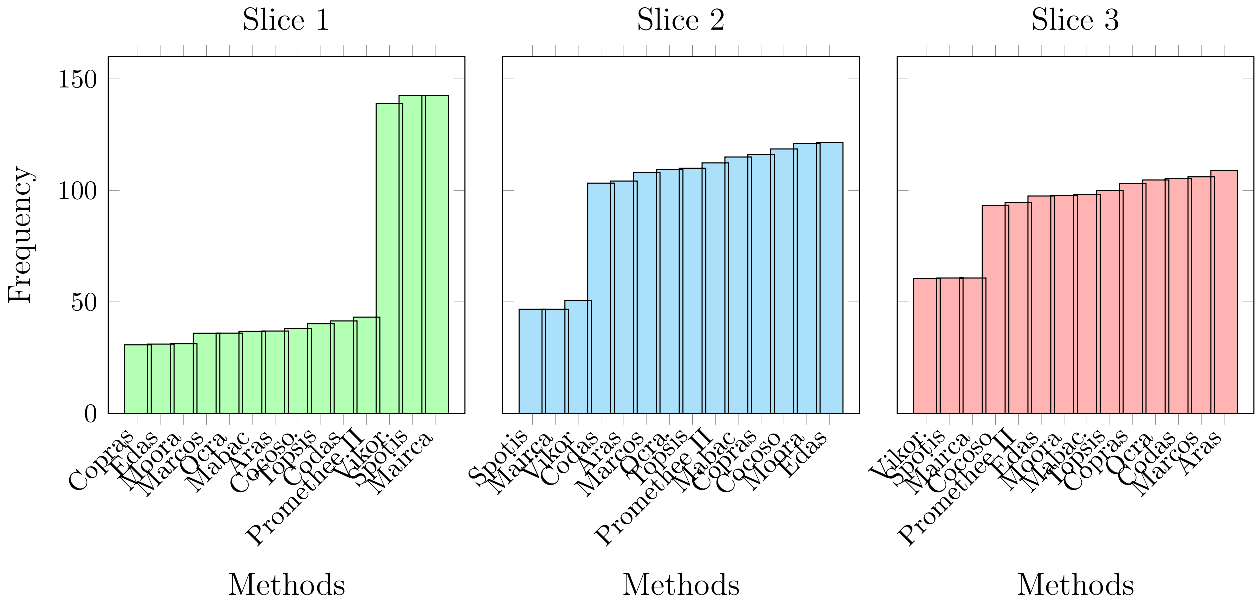
In both experiments, a linear regression model was used to adjust the response variable data, and then the result was summarized in an ANOVA, so that the multiple comparison using the Tukey's test could be done. All tests performed considered a confidence level of 95%.

6.4.1 Experiment 1

The results from Experiment 1 are reported in Table 12 and summarized in Figure 35. For all slices, there was rejection of the null hypothesis, i.e., the premisses of normality and independence of the residuals, in addition to the fact that the homoscedasticity of the variances were violated, according to equations (6.2), (6.3) and (6.4).

The performance of all methods for each slices is shown in Figure 36. For Slice 1, it appears that the Cocoso and Ocra methods had the lowest IQR, and this is confirmed in the KRUSKAL-WALLIS test. In general, most of the methods have a similar average behavior, with an exception for the Mairca, Spotis and Vikor ones, which have repeatedly shown difficulty in operating in environments with great variability in the selection criteria,

Figure 35 – Behavior of the NSSF DAF methods - TS 03: Experiment 1



Source: elaborated by the author.

and, for this reason, their results always diverge from the other MCDM methods. However, the analysis of the estimated coefficients of the regression model shows the superiority of the Spotis (5.96), Mairca (5.96) and Vikor (5.78) methods in the selection of Slice 1.

From the Tukey's test, it was observed that there was no significant difference between the methods in 55 comparisons out of 182 possible. This behavior is reported in the network graph of Figure 37a, where the formation of two clusters can be observed: the first formed by the Mairca, Spotis and Vikor methods, which denote the behavior of the descriptive analysis provided in the boxplot of Figure 36a; the second formed by the other methods with the leadership of the Promethee II method, which presents a slightly superior performance to the others.

For Slice 2, which is the slice with the greatest technical capacity, as specified in Table 5 and 6, the methods have a similar average behavior, showing only a greater difference in the variability of the response variable; however, it is necessary to verification of the main hypothesis defined in the equation (6.1). The Mairca, Vikor and Spotis methods presented a performance that diverges again from the others, as illustrated in Figure 36b. It is also observed that the Vikor and Cocoso methods had the lowest IQR. From the KRUSKAL-WALLIS test it was verified that there are differences between the methods, leading to the rejection of the null hypothesis of the equation (6.1). From the estimated coefficients analysis of the regression model there is evidence of a slight superiority of the methods Edas (0.85), Moora (0.83), and Cocoso (0.79), compared to the other ones. For

Table 12 – Methods Results: TS03 - Experiment 1

Slice	Methods	Mean	STD	Var	CI
Slice 1	Aras	36.91	14.63	214.15	31.72 – 42.09
	Cocoso	38.12	5.05	25.55	36.33 – 39.91
	Codas	41.45	20.41	416.63	34.22 – 49.69
	Copras	30.73	14.70	216.08	25.52 – 35.94
	Edas	31.06	13.77	189.68	26.18 – 35.94
	Mabac	36.82	11.24	126.40	32.83 – 40.80
	Mairca	142.61	18.82	354.12	135.93 – 149.28
	Marcos	35.94	14.79	218.87	30.69 – 41.19
	Moora	31.21	14.18	200.98	26.19 – 36.24
	Ocra	35.97	7.82	61.22	33.20 – 38.74
	Promethee II	43.15	12.41	154.07	38.75 – 47.55
	Spotis	142.61	18.82	354.12	135.93 – 149.28
	Topsis	40.18	18.78	352.65	33.52 – 46.84
	Vikor	138.88	25.41	645.73	129.87 – 147.89
Slice 2	Aras	104.18	32.96	1086.53	92.49 – 115.87
	Cocoso	118.58	11.30	127.63	114.57 -122.58
	Codas	103.24	41.55	1726.31	88.51 – 117.98
	Copras	116.12	30.26	915.48	105.39 – 126.85
	Edas	121.45	29.53	871.76	110.99 – 131.92
	Mabac	114.97	24.46	598.47	106.30 – 123.64
	Mairca	46.70	16.99	288.59	40.67 – 52.72
	Marcos	107.97	32.48	1054.66	96.45 – 119.48
	Moora	121.00	29.89	893.31	110.40 – 131.60
	Ocra	109.36	18.54	343.61	102.79 – 115.94
	Promethee II	112.33	22.17	491.35	104.47 – 120.19
	Spotis	46.70	16.99	288.59	40.67 – 52.72
	Topsis	109.94	34.46	1187.25	97.72 – 122.16
	Vikor	50.58	20.11	404.50	43.44 – 57.71
Slice 3	Aras	108.91	33.71	1136.59	96.95 – 120.86
	Cocoso	93.30	12.01	144.28	89.04 – 97.56
	Codas	105.30	42.80	1831.91	90.13 – 120.48
	Copras	103.15	30.75	945.45	92.25 – 114.05
	Edas	97.48	29.46	868.13	87.04 – 107.93
	Mabac	98.21	23.74	563.36	89.80 – 106.63
	Mairca	60.70	20.93	438.09	53.28 – 68.11
	Marcos	106.09	33.75	1139.21	94.12 – 118.06
	Moora	97.79	29.49	869.80	87.33 – 108.25
	Ocra	104.67	18.40	338.67	98.14 – 111.19
	Promethee II	94.52	21.53	463.57	86.88 – 102.15
	Spotis	60.70	20.93	438.09	53.28 – 68.12
	Topsis	99.88	36.00	1296.36	87.11 – 112.65
	Vikor	60.55	28.61	818.51	50.40 – 70.69

Legend: STD: Standard Deviation; Var: Variance ; CI: Confidence Intervals.

Source: elaborated by the author.

the Tukey test, it was found that there is no significant difference between the methods in 57 comparisons, which explains the perceived homogeneity in the Figure 37b.

In Slice 3, from analyzing the estimated coefficients of the regression model, it was observed that basically all methods had similar performance, as can be seen in Figure 36c. However, the Mairca, Spotis and Vikor methods had greater difficulties in evaluating the traffic conditions to Slice 3. This slice has a moderate capacity compared to the other slices and aims to serve a specific niche in the mMTC scenario. From the Tukey's test, it was observed that there is no difference between the methods for 57 comparisons, as seen in Figure 37c.

The distribution adjustment of the response variable (Frequency of Selections) was considered using a binomial distribution, and the $sqrt()$ function was applied to approximate a normal distribution for all slices. These adjustments are illustrated in Figures 38, 39 and 40.

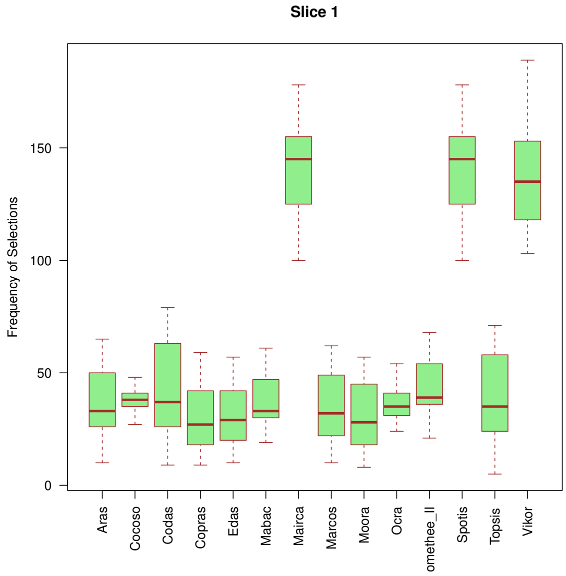
The findings presented in this section are further complemented by the results shown in Appendix B (TS03: Experiment 2 and priority data). The experiments shown in Appendix B differ from those presented in this section by their weight setup. In addition to validating the behavior of the proposed hybrid algorithms and the efficiency of the NSSF DAF architecture, the occurrence of 200 priority flows was also considered.

6.5 Chapter summary

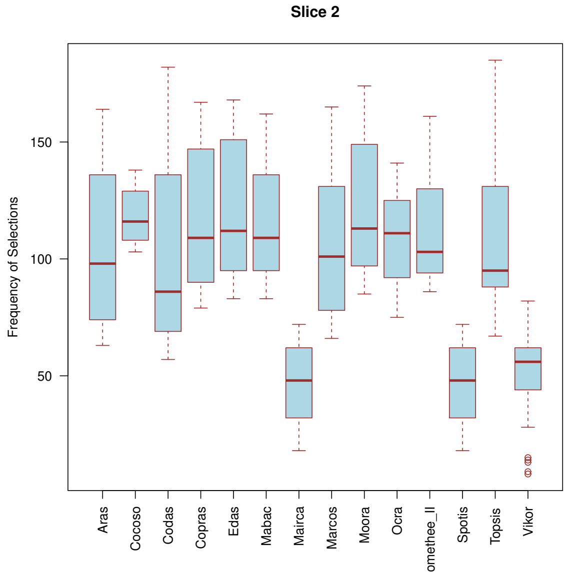
In this chapter, the results and analysis for the test scenarios defined in Chapter 5 were presented. It was observed, from the results obtained and the statistical analyzes measured, that the proposed framework is efficient for the evaluated scenarios.

For the experiments in the TS-01 and T-03 scenarios (see Appendix B for further details), there was a need for verification and validation of the response variable, which, although it presented characteristics of a normal distribution, when different statistical tests were applied, it was not possible to obtain models that represented the data with reliability, as well as it made performance comparisons between methods difficult. For example, for TS-01, considering Experiment 1 and Slice 1, if the distribution followed a binomial distribution, the likelihood of the Ocra method selecting Slice 1 would be 3.7 to 4.5 times greater than another method, considering a level of 95% confidence. However, such comparisons were not possible as demonstrated in the simulated envelope graphs. In this way, it was decided to use a linear regression model applying the correction provided

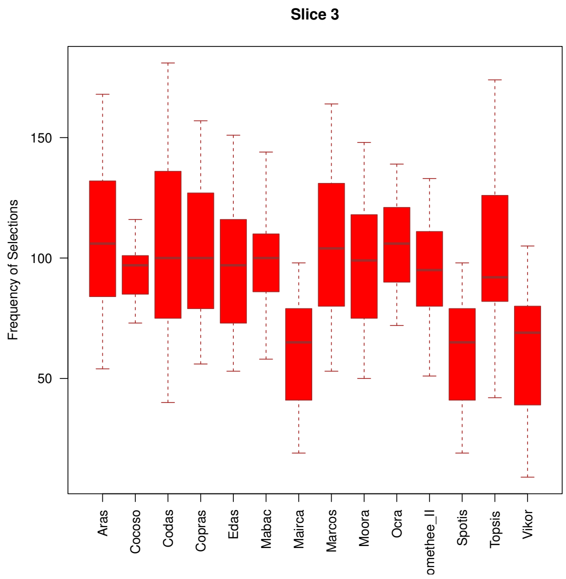
Figure 36 – Descriptive analysis using boxplot - TS 03: Experiment 1.



(a) Slice 1



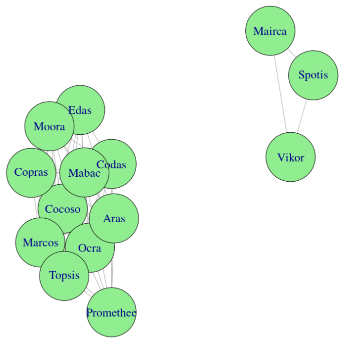
(b) Slice 2



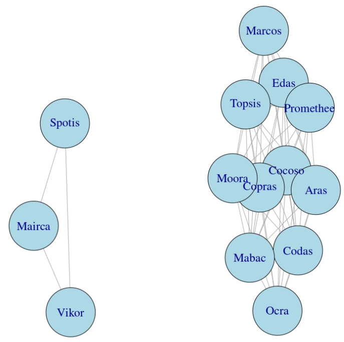
(c) Slice 3

Source: elaborated by the author.

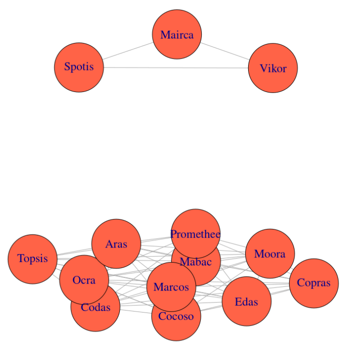
Figure 37 – Network graph TS03: Experiment 1. There is no significant difference between this methods.



(a) Slice 1



(b) Slice 2

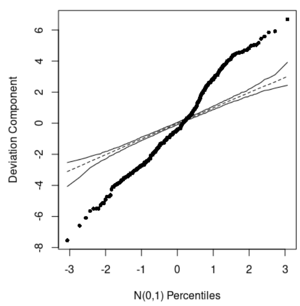


(c) Slice 3

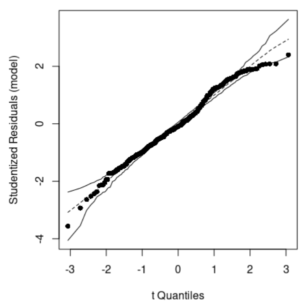
Legend: 95% confidence intervals comparing each pair of methods.

Source: elaborated by the author.

Figure 38 – Simulated envelope TS03: Experiment 1 - Slice 1.



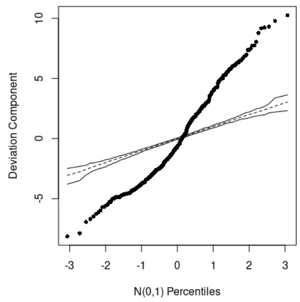
(a) Binomial GLM.



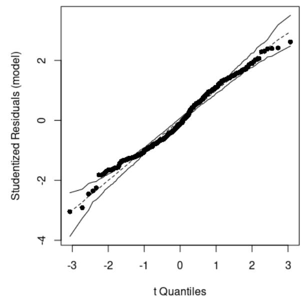
(b) Normal LM.

Source: elaborated by the author.

Figure 39 – Simulated envelope TS03: Experiment 1 - Slice 2.



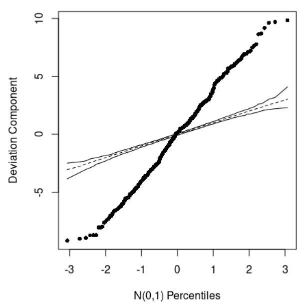
(a) Binomial GLM.



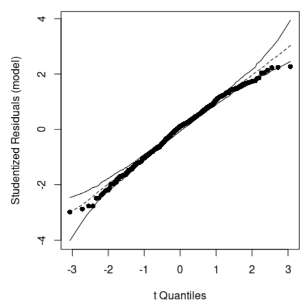
(b) Normal LM.

Source: elaborated by the author.

Figure 40 – Simulated envelope TS03: Experiment 1 - Slice 3.



(a) Binomial GLM.



(b) Normal LM.

Source: elaborated by the author.

by the *sqrt*() function. This strategy approximated the distribution to a normal one, allowing the use of an ANOVA model and the comparison of methods MCDM based on the use of multiple comparison methods.

For the TS-01, it was verified, in all experiments, that the Ocra method presented the best results for Slice 1 (Remote Driving), and this demonstrates the efficiency of the method in situations where the traffic requirements were more critical, i.e., that require low latency and loss and high reliability. For Slice 2, the Cocoso and Promethee II methods presented very consistent and regular results throughout all experiments, demonstrating greater adherence to traffic where milder attenuation's and more homogeneous characteristics of the criteria existed. For Slice 3, there was a dissonant behavior of the Spotis, Mairca and Vikor methods. These methods showed great difficulty in dealing with scenarios of great variability in the decision criteria. In addition, such methods had difficulty in dealing with network traffic with high QoS requirements (the case of Slice 1), which is specified to meet "Remote Driving", considered one of the most critical scenarios for 5G and future networks.

For the TS-02, it was observed that the use of fuzzy logic, in conjunction with the MCDM methods, allowed a performance gain of accuracy in the slices selection. In addition, it enabled the criteria to have a normal distribution, which facilitated the use of parametric tests, and a less laborious statistical inference. However, the definition of the set of rules requires the knowledge of an expert, which makes its application difficult in production environments that are based on fully automated processes, such as those that integrate MLOps models.

For TS-03 and TS-03 Alarm (priority data), the methods showed similar performance in all experiments, with a more homogeneous distribution of the response variable. The Spotis, Mairca and Vikor methods were dissonant in relation to the other methods for all experiments. These methods prioritized Slice 1 almost exclusively. In experiments with a fair distribution of weights, there is less variability in the response variable, which implies a smaller number of outliers. In this sense, residual plots were used to identify the outliers.

Overall, in the experiments conducted in the TS-03 and TS-03 Alarm scenarios, for Slice 2, the Edas and Moora methods showed greater selection accuracy, and for Slice 3, the Copras and Mabac methods were better in almost all experiments. The NSSF DAF can be configured according to the purpose of previously instanced slices, helping in the network orchestration process.

7 CONCLUSIONS

This work presented a framework proposal that implements the NSSF function for 5G and future networks, using an approach that employs the concepts of Intelligent Edge Computing, MLOps and AIOps in cloud native environments. In addition, a comparative study between some slice selection strategies are presented. Three test setup environments with different implementation technologies were used, seeking to assess the flexibility and adaptability of the solution for different scenarios.

In general, each scenario includes 3 vertical slices in a multi-domain environment, aiming at three common services in the context of 5G networks. In addition, we used a strategy from the K-means method to group the UE data flows, so as to characterize the available slices and, subsequently, define the weights that would be used in the MCDM methods, whose purpose consisted of defining and selecting the best slice available to receive the network flow during operating time.

Initially, it was necessary to study the slice selection models available in the literature, as well as verifying the limitations of the works related, identifying important research gaps. Next, the techniques were verified in order to solve the problem, so as to support the choice and option for the methods covered herein that constitute the core of the NSSF DAF framework.

The MCDM methods proved to be favorable, due to their simplicity and efficiency when compared to other strategies; they did not require a history of virtual network operations and slices, which can be very useful for newly-created slices. To define the weights and priorities between the criteria, the K-means clustering algorithm was considered, providing the opportunity for further studies and to adopt hybrid methods. Furthermore, the proposed approach is extensible and allows the addition of new slice selection strategies, considering the possibility of exploring reinforcement learning, evolutionary algorithms, Bayesian networks, or even networks recurrent along with the MCDM methods in future works.

Considering the restrictions of the scenario, and from the analysis of the results, it was verified that the proposed solution proved to be promising, as it can be transparently deployed in the core or at the edge of networks that are already in operation. The architecture of the framework, as well as its implementation models, allow the performance of new tests considering other scenarios, so that improvements to the algorithms developed may be proposed, as well as other techniques, strategies, and methods for the core of the proposed framework can be evaluated.

It also observed that there are significant differences between the evaluated methods. In this way, using the MLOps concept, it is possible to i) scale, load and retrain models, 2) adjust parameters, criteria weights and preference functions in network runtime, iii) optimize the use of slices in test environments, and iv) direct data flows according to the agreed SLA's. Furthermore, it is possible to evaluate the adoption of data grouping models, aiming to select the most suitable slice for the user considering the traffic generated. Hence, this work proposed an optimized E2E approach for the traffic management of access networks to the 5G Core.

Finally, we concluded that selecting slices is still an open problem, which invites researchers to study and develop new techniques, approaches, and solutions. Our proposed framework provides compatibility with the current specification standards, thus constituting a promising solution for 5G and future networks in the context of NS.

The NSSF DAF solution can also be applied in several application niches, such as situations that need to integrate large amounts of data from devices linked to the IoT context to cloud services, in addition to the application in specialized slices such as eMBB, mMTC and URLLC.

7.1 Answers for the research questions

This subsection contains the answers to the main research question (4.1.1) and the secondary questions (4.1.2).

7.1.1 Main research questions

To answer the five RQs, an approach that employs machine learning, fuzzy logic and decision-making methods called NSSF DAF was implemented. The proposed approach focuses on the evaluation and dynamic mapping of the appropriate QoS requirements for each type of service, user profile and specialized slice, initially focusing on the vertical model from edge computing, and then providing a distributed solution in the model horizontal based on 3GPP OpenAPIs, MLOps and CNFs.

Different testbeds were conducted to validate the proposed solution, as well as to evaluate its efficiency and ability to integrate with the specifications defined by the standardization bodies. The combination of these insights resulted in a scalable, extensible and promising approach to provisioning NSSF service in 5G and future networks.

- **RQ1. "How to select the best slice?"**.

  Different solutions have been reported in the literature, with different approaches. A portion of these solutions discussed in Section 3.1 have been validated in simulation models and focus on the RAN. However, although these solutions point to interesting ways, the biggest challenge focuses on proposing solutions in a viable deployment model in networks that are in operation, and that includes an E2E network slicing architecture, considering the preferences and the environment where the UE is inserted. The framework proposed in this thesis covers all these gaps. Evidence can be found in Chapters 4 and 5.

- **RQ2. "Do the selected slices provide the necessary requirements for the user?"**.

  Dynamic mapping of user preferences to incoming traffic requires integration between different technologies and methods. In this sense, strategies for sharing preferences between the UEs and the edge need to be done in a transparent and efficient way. Furthermore, it is expected that the selection slices can support the creation and/or adjustment of slices previously instantiated by network orchestrators. The solution implemented and validated in this work selects the best slice according to user preferences and traffic, and provides the data model required from network orchestrators. Evidence can be found in Sections 4.1.2 and 4.1.3.

- **RQ3. "Are the selection criteria modeled in a generic way, independently of the access networking technology?"**

  In general, market solutions usually work as a black box and have specialist systems for analyzing and forwarding packets based on specific marking standards (e.g. MPLS, SR-IPv6, EVPN). These solutions require the acquisition of numerous proprietary appliances that aim to provide a complete pipeline, for example, RAN, Edge and Core. The approach employed in this work makes use of open sources NFVs that implement specific applications for data processing directly from the TCP/IP, SCTP and NGAP protocols stack. In this way, the QoS metrics used in the NSSF function are obtained regardless of the RAN technology (All access modes), aggregation and core networks, not requiring changes in legacy networks. The evidence are detailed in Sections 4.1, 5.2.1, and 5.2.3.

- **RQ4. "Do the approaches employ techniques aimed at the integration and interoperability between RAN, Edge, and Core networks?"**.

  Cloud native networks (e.g. 5G and 6G) consider that the entire network slicing architecture must be virtualized. In this way, the NFs that can be instantiated in the RAN, Edge or Core must offer and consume services through APIs. The specifications of these APIs should provide a set of operations comprising interfaces (SBI) to discover and identify available services and repositories, in addition to allowing the creation of exposure models for new functions (e.g. NEFs). These requirements have united the efforts of different standardization organizations, in particular 3GPP through the 3GPP TS 29.501 specification and its updates, which recommend that APIs should be designed as RESTful and allow JSON and YAML data models.

  The framework proposed in this work implements these models and enables integration and interoperability between NFs that can be deployed anywhere. For this, it uses cloud orchestration technologies (K8s Clusters) and NFV/SDN functions. The evidence are detailed in Section 4.1.3, and in the testbeds performed in Chapter 5.

- **RQ5. "Do the solutions provide compatibility with the main specifications standards under development?"**.

  To allow integration with different network orchestration tools (e.g. OSM, ONAP, EMCO), as well as open 5G Core implementations (e.g. Open5GS and Free5GC[1]), it is necessary that slice selection solutions i) use the 3GPP OpenAPIs specifications, ii) follow compliance policies defined by ETSI for edge applications, and iii) allow the integration of data from the Open RAN ecosystem and respecting the QoS Regulation Manual defined by the ITU-T.

  Despite the limitations of computational resources (NFVI) used in this experiments, all slices instantiated in the test setups conducted in this work followed the technical specifications of these organizations. In this way, the NSSF DAF was validated following the guidelines. The evidence can be evaluated in Chapter 5.

[1]https://free5gc.org/

7.1.2 Secondary research questions

- **SRQ1: "Is it feasible to propose a mathematical model that considers the features and behavior of the different QoS variables for the NSSF function?"**.

  The modeling of the slice selection problem requires the evaluation of numerous variables, most of which have conflicting objectives. Thus, it is necessary to formulate algorithms for these multi-objective problems. In general, it is necessary to minimize or maximize different variables, model the numerous restrictions, adjust and readjust the parameters, among others (WANG; PEI; LI, 2023). The biggest challenge is to do all the tasks and steps in network runtime. In addition, the formulation of objective functions and the obtainment of optimal solutions are also an arduous task, given the variability of the input variables, i.e., these variables do not always have well-defined limits and may undergo sudden changes in short time intervals.

  For the experiments performed in this work, 07 to 08 QoS variables were used. Proposing solutions on this scale involves the application of multi-objective optimization methods, in particular, the application of evolutionary multi-objective algorithms (MOEA) (YACOUBI et al., 2023), since classical optimization algorithms do not deal with the problem. This task is complex and requires a steep convergence time.

  The solution adopted in this work used hybrid algorithms of machine learning and decision-making methods.

- **SRQ2:"Are the use of artificial intelligence techniques sufficient to predict slice behavior during the network runtime?"**.

  The experiments performed in this work showed that machine learning and fuzzy logic techniques show promising results; however, it was noticed the need for the refinement of the provided solutions. In this sense, the decision-making methods played a fundamental role, since they do not require traffic history, and assemble their matrices in network runtime. However, these strategies and methods will be ineffective if there is not a solid methodology for data collection, training, re-training and model adjustments in a continuous cycle. In this way, the application of the MLOps concept proves to be fundamental, correcting the lag of the models and providing continuous improvement.

  The solution proposed in this work used all these approaches.

- **SRQ3: "How to ensure that the selection of slices made by a given user equipment is carried end-to-end? What granularity of automation and interoperability is required to orchestrate multiple network elements across different administrative domains?"**.

  This question involves different aspects. Initially, it is necessary to consider technical and operational issues, and then issues involving regulatory and business models. Ensuring slice selection involves different network entities, requiring adaptation of the following points: i) inter-domain slices must be available for the NSSF function; ii) it is necessary to guarantee that the selection of slices at the user and edge levels are maintained in inter-domain scenarios through SLAs; iii) SLAs regulation and continuity models are needed through a federated orchestration service; iv) mobility management protocols and specialized network functions are needed to guarantee E2E service; v) efficient and intelligent MLOps strategies are needed to enable operations without human intervention; vi) business models that support these operations are needed.

  All these issues constitute open research problems and demand solutions. For the approach proposed in this work, if the orchestrator is multi-domain, the NSSF DAF will also be. However, the framework can act in local, regional and federalized domains. The only requirement concerns slice visibility/availability for the NSSF cloud native function.

7.2 Recommendations for future work

The solution proposed in this research project constitutes an important part of scenarios involving future networks. In general, the continuity of this work can be divided into three major blocks, namely: **i) Block 1** - definition, analysis and testing of new approaches for the proposed framework core; **ii) Block 2** - improvements to adaptability mechanisms and updating of implementations and deployment models; **iii) Block 3** - development and extension of functionalities to facilitate the resolution of open research problems.

- **Block 1:** An interesting path for further work consists of evaluating and comparing other approaches to the *Processor* module of the NSSF DAF framework. Studies involving: Deep Learning; SVM; Bayesian Networks; Decision Trees, and MOEA. The idea is to verify the applicability of these methods and techniques from the perspective of chaotic systems, i.e., those that present: uncertain behavior; nonlinear systems; non-regulation; orderliness and disorder.

Although several testbeds have been conducted to validate the proposal in this work, other possible ways could involve validation through formal methods, such as: Hierarchical Colored Petri Nets (HVL); Stochastic Petri Nets (SPN), Markov Chains; BAN logic; and Applied pi Calculus.

- **Block 2:** A very relevant research contribution would be the implementation of intelligents triggers along the chain of NFVs and VNFs. These triggers would implement an intelligent scheduler of methods and techniques for the core of the framework, according to traffic demand and utilization rates. Although, the solution presented in this work implements the concepts of DevOps and MLOps, the triggers that signal the framework operations follow a Jenkins pipeline, and the operations of the controller-manager (kube and cloud) of the K8s orchestrator. The proposition of intelligent models that dynamically read the scenario from multi-agent systems or intelligent bots would provide greater flexibility and scalability of the NFs, especially the 3GPP NWDAF[2] function (TS 29.520). Projects like Jina[3] AI, which implement multimodal pipelines with cloud native technologies, can be valuable in these researches.

- **Block 3:** The possibilities for future work defined in this block aim to give the NSSF DAF framework or another solution that implements the 5GC NSSF function ways to solve open problems from the research questions and market. In this sense, different modules or extensions can be added to the solution using the same integration model based on 3GPP OpenAPIs.

  Thus, we can highlight the need for adding the following modules:

  1. Mobility management module. After the process of selecting slices, and indicating the best slice for the orchestrator, it is necessary to implement specialized NFs to guarantee the continuity of the service in the handover process.
  2. 6G Extension Module. Aims to meet the KPI's specifications defined by the NGNM for 6G networks. These specifications present specific requirements for machine learning and deep learning models in a MLOps approach.
  3. Integration module for proposing slice selection models as a federated service.

[2]https://www.3gpp.org/DynaReport/29520.htm
[3]https://github.com/jina-ai/jina

4. Extension of the NSSF DAF framework to user (data) planes, through the P4[4] programming language and DPDK. Need to evaluate the performance of the module to analyze the technical feasibility of deployment in switches layer 2 and 3.

7.3 Other Results

This doctoral research resulted in scientific publications and open source artifacts. In the following subsections, we present these results.

7.3.1 Published Papers

- da Silva, D.C.; de Sousa, M.A.F.; Oliveira, W.; Barbosa, A.; Henrique, P.R.; Bressan, G.; Silveira, R.M.; Prasad. R. **NSSF function in 6G networks based on MLOps deployment model**. CONASENSE: An Interdisciplinary Vision Towards 6G: Comunication, Navigation, Sensing and Services. IEEE WPMC 2023. Orlando, Florida, USA, 2023. **Under review**.

- da Silva, D.C.; de Sousa, M.A.F.; Bressan, G.; Silveira, R.M. **A benchmark analysis of machine learning and decision-making methods for NSSF function in a cloud-native 5G network**. Computers and Electrical Engineering. 2023. Qualis A2. **Under review**.

- da Silva, D.C.; Batista, J.O.R., Jr.; de Sousa, M.A.F.; Mostaço, G.M.; Monteiro, C.d.C.; Bressan, G.; Cugnasca , C.E.; Silveira, R.M. **A Novel Approach to Multi-Provider Network Slice Selector for 5G and Future Communication Systems**. Sensors 2022, 22, 6066. https://doi.org/10.3390/s22166066 . Qualis A1.

- da Silva, D.C.; de Sousa, M.A.F.; Bressan, G.; Silveira, R.M. **5G Network Slice Selector in IoT Services Scenarios with QoS Requirements Guarantee**. 2022 IEEE Wireless Communications and Networking Conference (WCNC), Austin, TX, USA, 2022, pp. 90-95, doi: 10.1109/WCNC51071.2022.9771744.

- Batista, J.O.R., Jr.; da Silva, D.C.; Martucci, M., Jr.; Silveira, R.M.; Cugnasca, C.E. **A Multi-Provider End-to-End Dynamic Orchestration Architecture**

[4] https://opennetworking.org/p4/

Approach for 5G and Future Communication Systems. Appl. Sci. 2021, 11, 11914. https://doi.org/10.3390/app112411914

7.3.2 Developed Software

- NSSF DAF: Network Slice Selection Function Decision-Aid Framework[5].

7.4 Final remarks

This section concludes the research. As it was described, the main contributions delivered was performed, as well as the answers to the research questions listed in Chapter 1. Opportunities for continuing the research in future works were also considered, divided and detailed into three large blocks.

In general, the solution proposed in this work concentrated efforts on proposing an applicable solution for the slice selection problem in cloud-native networks. The architecture of the developed framework adhered to the specifications of the main standards organizations. In addition, some premises were assumed, such as: the solution's deployment model should be applicable to the main cloud virtualization and orchestration tools; the integration model should be transparent in networks that were already in operation; the solution should be scalable and extensible, allowing the integration of new modules; the data collection model should be minimally invasive, not generating overhead on the network; it should be independent of specific hardware manufacturers and models; it should propose continuous life cycle improvement of the VNFs; and use preferentially open source software.

Finally, we believe that the work achieved the outlined objectives, ensuring the delivery of the expected contributions.

[5]Available in https://github.com/gprisa/nssfdaf

REFERENCES

3GPP. TS 23.207 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; End-to-end Quality of Service (QoS) concept and architecture (Release 16). v. 16.0.0, p. 0–39, 2020. Disponível em: <https://www.3gpp.org/ftp/Specs/archive/23_{series}/23.207/>.

3GPP. **Service requirements for the 5G system**. [S.l.], 2021. Version 18.4.0.

3GPP. **3GPP A Global Initiave**. 2023. Available online: <https://www.3gpp.org> (accessed on 11 07 2023).

AFAQ, M. et al. Towards 5G network slicing for vehicular ad-hoc networks: An end-to-end approach. **Computer Communications**, Elsevier B.V., v. 149, n. July 2019, p. 252–258, 2020. ISSN 1873703X. Disponível em: <https://doi.org/10.1016/j.comcom.2019.10.018>.

ALINEZHAD, A.; KHALILI, J. Copras method. In: _____. **New Methods and Applications in Multiple Attribute Decision Making (MADM)**. Cham: Springer International Publishing, 2019. p. 87–91. ISBN 978-3-030-15009-9. Disponível em: <https://doi.org/10.1007/978-3-030-15009-9_12>.

AUMAYR, E. et al. Service-based analytics for 5g open experimentation platforms. **Computer Networks**, v. 205, p. 108740, 2022. ISSN 1389-1286. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1389128621005892>.

BABASHAMSI, P. et al. Integrated fuzzy analytic hierarchy process and VIKOR method in the prioritization of pavement maintenance activities. **International Journal of Pavement Research and Technology**, v. 9, n. 2, p. 112–120, 2016. ISSN 1996-6814. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1996681416000225>.

BAKMAZ, B.; BOJKOVIC, Z.; BAKMAZ, M. TOPSIS-based approach for network slice selection in 5G mobile systems. **International Journal of Communication Systems**, v. 33, n. 11, p. 1–14, 2020. ISSN 10991131.

BARAKABITZE, A. A. et al. 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. **Computer Networks**, Elsevier B.V., v. 167, 2020. ISSN 13891286.

BARLACCHI, G. et al. A multi-source dataset of urban life in the city of Milan and the Province of Trentino. **Scientific Data**, v. 2, n. 1, p. 150055, 2015. ISSN 2052-4463. Disponível em: <https://doi.org/10.1038/sdata.2015.55>.

BARRACHINA-MUñOZ, S.; PAYARó, M.; MANGUES-BAFALLUY, J. Cloud-native 5g experimental platform with over-the-air transmissions and end-to-end monitoring. In: **2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)**. [S.l.: s.n.], p. 692–697, 2022.

BELTRE, A.; SAHA, P.; GOVINDARAJU, M. Framework for analysing a policy-driven multi-tenant kubernetes environment. In: **2021 IEEE Cloud Summit (Cloud Summit)**. [S.l.: s.n.], p. 49–56, 2021.

BOJKOVIC, Z. S.; BAKMAZ, B. M.; BAKMAZ, M. R. Influences of Weighting Techniques on TOPSIS-based Network Slice Selection Function. In: **2019 14th International Conference on Advanced Technologies, Systems and Services in Telecommunications, TELSIKS 2019 - Proceedings**. [S.l.: s.n.], p. 270–277, 2019. ISBN 9781728108780.

BONATI, L. et al. Open, Programmable, and Virtualized 5G Networks: State-of-the-Art and the Road Ahead. **Computer Networks**, Elsevier B.V., v. 182, n. July, p. 107516, 2020. ISSN 13891286. Disponível em: <https://doi.org/10.1016/j.comnet.2020.107516>.

BU, C. et al. Routing as a service (RaaS): An open framework for customizing routing services. **Journal of Network and Computer Applications**, Elsevier Ltd, v. 125, n. June 2018, p. 130–145, 2019. ISSN 10958592. Disponível em: <https://doi.org/10.1016/j.jnca.2018.10.010>.

CAMPOLO, C. et al. Slicing on the road: Enabling the automotive vertical through 5G network softwarization. **Sensors (Switzerland)**, v. 18, n. 12, p. 1–17, 2018. ISSN 14248220.

CASTILLO, O.; CASTRO, J.; MELIN, P. Type-2 fuzzy logic systems. In: _____. [S.l.: s.n.], 2022. p. 5–11. ISBN 978-3-030-96514-3.

CHAHAL, M.; HARIT, S. Network selection and data dissemination in heterogeneous software-defined vehicular network. **Computer Networks**, Elsevier B.V., v. 161, p. 32–44, 2019. ISSN 13891286.

CHEN, W. K. et al. Optimal Network Slicing for Service-Oriented Networks with Flexible Routing and Guaranteed E2E Latency. **arXiv**, p. 1–14, 2020. ISSN 23318422.

CHEN, X.; ZHAO, Y.; LI, Y. QoE-Aware wireless video communications for emotion-aware intelligent systems: A multi-layered collaboration approach. **Information Fusion**, v. 47, n. June 2018, p. 1–9, 2019. ISSN 15662535.

CHIANG, M.; ZHANG, T. Fog and IoT: An Overview of Research Opportunities. **IEEE Internet of Things Journal**, IEEE, v. 3, n. 6, p. 854–864, 2016. ISSN 23274662.

CHOI, Y. I.; PARK, N. Slice architecture for 5G core network. In: **International Conference on Ubiquitous and Future Networks, ICUFN**. [S.l.: s.n.], p. 571–575, 2017. ISBN 9781509047499. ISSN 21658536.

CHOUDHARI, C. S.; PATIL, R.; SARAF, S. Deployment of 5g core for 5g private networks. In: **2022 International Conference on Industry 4.0 Technology (I4Tech)**. [S.l.: s.n.], p. 1–6, 2022.

CISCO. **About Cisco**. 2022. Available online: <https://www.cisco.com/c/en/us/about.html> (accessed on 21 05 2022).

CONDOLUCI, M. et al. 5G V2X system-level architecture of 5GCAR project. **Future Internet**, v. 11, n. 10, 2019. ISSN 19995903.

CONDOLUCI, M.; MAHMOODI, T. Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges. **Computer Networks**, Elsevier B.V., v. 146, p. 65–84, 2018. ISSN 13891286. Disponível em: <https://doi.org/10.1016/j.comnet.2018.09.005>.

CUI, Y. et al. Machine Learning based Resource Allocation Strategy for Network Slicing in Vehicular Networks. In: **2020 IEEE/CIC International Conference on Communications in China (ICCC)**. [S.l.: s.n.], p. 454–459, 2020.

DEZERT, J. et al. The SPOTIS rank reversal free method for multi-criteria decision-making support. **Proceedings of 2020 23rd International Conference on Information Fusion, FUSION 2020**, 2020.

DIMOLITSAS, I. A Multi-Criteria Decision Making Method for Network Slice Edge Infrastructure Selection. In: **Proceedings of 3rd International Workshop on Advances in Slicing for Softwarized Infrastructures**. [S.l.]: IEEE, p. 1–7, 2020. ISBN 9781728156842.

DOGRA, A.; JHA, R. K.; JAIN, S. A Survey on beyond 5G network with the advent of 6G: Architecture and Emerging Technologies. **IEEE Access**, p. 1, 2020.

ETSI. **Developing Software for Multi-Access Edge Computing**. [S.l.: s.n.], 2019. ISBN 9791092620290.

ETSI. 5G; Service requirements for enhanced V2X scenarios (3GPP TS 22.186 version 16.2.0 Release 16). **System**, v. 16.2.0, p. 1–16, 2020.

ETSI. 5G; Service requirements for the 5G system (3GPP TS 22.261 version 16.14.0 Release 16). **System**, v. 16.14.0, p. 1–72, 2021.

ETSI. **European Standards Organization**. 2023. Available online: <https://www.etsi.org> (accessed on 11 07 2023).

FAROOQUI, M. N. I. et al. A comparative analysis of 5g network slicing simulators. In: **2022 3rd International Conference on Innovations in Computer Science Software Engineering (ICONICS)**. [S.l.: s.n.], p. 1–8, 2022.

FIGUEIRA, J.; GRECO, S.; EHRGOTT, M. **Multiple Criteria Decision Analysis: State of The Art Surveys**. [S.l.]: Springer Science, 2005. v. 78. (International Series in Operations Research & Management Science, v. 78).

GANKIDI, N. et al. Customer segmentation using machine learning. In: **2022 2nd International Conference on Intelligent Technologies (CONIT)**. [S.l.: s.n.], p. 1–5, 2022.

GLIGOROSKI, D.; KRALEVSKA, K. Expanded combinatorial designs as tool to model network slicing in 5G. **IEEE Access**, IEEE, v. 7, p. 54879–54887, 2019. ISSN 21693536.

GUAN, W.; ZHANG, H.; LEUNG, V. C. M. Analysis of Traffic Performance on Network Slicing Using Complex Network Theory. **IEEE Transactions on Vehicular Technology**, v. 69, n. 12, p. 15188–15199, 2020.

HABIBI, M. A.; HAN, B.; SCHOTTEN, H. D. Network Slicing in 5G Mobile Communication Architecture, Profit Modeling, and Challenges. n. August, 2017. Disponível em: <http://arxiv.org/abs/1707.00852>.

HADOOP. **Apache Hadoop**. 2022. Available online: <https://hadoop.apache.org/> (accessed on 19 12 2022).

Haghi Kashani, M. et al. A systematic review of iot in healthcare: Applications, techniques, and trends. **Journal of Network and Computer Applications**, v. 192, p. 103164, 2021. ISSN 1084-8045. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804521001764>.

HEZER, S.; GELMEZ, E.; ÖZCEYLAN, E. Comparative analysis of TOPSIS, VIKOR and COPRAS methods for the COVID-19 Regional Safety Assessment. **Journal of Infection and Public Health**, v. 14, n. 6, p. 775–786, 2021. ISSN 1876-0341. Disponível em: <https://www.sciencedirect.com/science/article/pii/S187603412100068X>.

HUSAIN, S. et al. Mobile edge computing with network resource slicing for Internet-of-Things. **IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings**, v. 2018-Janua, n. December 2017, p. 1–6, 2018.

HUSSEIN, A. H. Internet of things (iot): Research challenges and future applications. **International Journal of Advanced Computer Science and Applications**, The Science and Information Organization, v. 10, n. 6, 2019. Disponível em: <http://dx.doi.org/10.14569/IJACSA.2019.0100611>.

IANNELLI, M.; HILL, M.; WANG, L. Applying Machine Learning to End-to-end Slice SLA Decomposition. In: **Proceedings of 3^{rd} International Workshop on Advances in Slicing for Softwarized Infrastructures**. [S.l.]: IEEE, p. 92–99, 2020. ISBN 9781728156842.

IETF. **Internet Engineering Task Force**. 2023. Available online: <https://www.ietf.org> (accessed on 11 07 2023).

JAIN, A.; LOPEZ-AGUILERA, E.; DEMIRKOL, I. Are mobility management solutions ready for 5G and beyond? **Computer Communications**, v. 161, p. 50–75, 2020. ISSN 0140-3664. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0140366419316032>.

JAMES, G. et al. **An Introduction to Statistical Learning: with Applications in R**. Springer US, 2021. (Springer Texts in Statistics). ISBN 9781071614181. Disponível em: <https://books.google.com.br/books?id=5dQ6EAAAQBAJ>.

JOŠILO, S.; DÁN, G. Joint wireless and edge computing resource management with dynamic network slice selection. **arXiv**, 2020. ISSN 23318422.

JR, J. O. R. B. et al. Distributing the Cloud Towards Autonomous Resilient 5G Networking. In: **ICTC 2019 - 10^{th} International Conference on ICT Convergence: Leading the Autonomous Future**. [S.l.]: IEEE, p. 854–859, 2019. ISBN 9781728108926.

JR, J. O. R. B. et al. Towards 5G Requirements: Performance Evaluation of a Simulated WSN Using SDN Technology. In: **12th EFITA (European Federation for Information Technology in Agriculture, Food and the Environment) HAICTA-WCCA Congress**. [s.n.], p. 24–29, 2019. ISBN 978-618-84798-0-7. Disponível em: <https://efita-org.eu/efita-2019/>.

JR., J. O. R. B. et al. A multi-provider end-to-end dynamic orchestration architecture approach for 5g and future communication systems. **Applied Sciences (Switzerland)**, v. 11, n. 24, 2021. ISSN 20763417.

KARATAS, F.; KORPEOGLU, I. Fog-Based Data Distribution Service (F-DAD) for Internet of Things (IoT) applications. **Future Generation Computer Systems**, v. 93, p. 156–169, 2019. ISSN 0167739X.

KHEDDAR, H. et al. An efficient model for horizontal slicing in 5g network using practical simulations. In: **2022 5th International Conference on Signal Processing and Information Security (ICSPIS)**. [S.l.: s.n.], p. 158–163, 2022.

KIM, D.; KIM, S. Network slicing as enablers for 5G services: state of the art and challenges for mobile industry. **Telecommunication Systems**, Springer US, v. 71, n. 3, p. 517–527, 2019. ISSN 15729451. Disponível em: <https://doi.org/10.1007/s11235-018-0525-2>.

KIM, Y.; KIM, S.; LIM, H. Reinforcement learning based resource management for network slicing. **Applied Sciences (Switzerland)**, v. 9, n. 11, 2019. ISSN 20763417.

KLIKS, A. et al. Perspectives for resource sharing in 5G networks. **Telecommunication Systems**, Springer US, v. 68, n. 4, p. 605–619, 2018. ISSN 15729451. Disponível em: <https://doi.org/10.1007/s11235-017-0411-3>.

KREUZBERGER, D.; KüHL, N.; HIRSCHL, S. Machine learning operations (mlops): Overview, definition, and architecture. **IEEE Access**, v. 11, p. 31866–31879, 2023.

KUBERNETES. **Kubernetes: Production-Grade Container Orchestration**. 2022. Available online: <https://kubernetes.io/> (accessed on 21 12 2022).

KUNDAKCı, N. A Comparative Analyze Based On EATWOS and OCRA Methods For Supplier Evaluation. **Alphanumeric Journal**, v. 7, n. 1, 2019.

LI, L.; OTA, K.; DONG, M. DeepNFV: A Lightweight Framework for Intelligent Edge Network Functions Virtualization. **IEEE Network**, v. 33, n. 1, p. 136–141, 2019.

LIU, L.-p. et al. Application of k-means ++ algorithm based on t-sne dimension reduction in transformer district clustering. In: **2020 Asia Energy and Electrical Engineering Symposium (AEEES)**. [S.l.: s.n.], p. 74–78, 2020.

LIU, Q.; HAN, T.; MOGES, E. EdgeSlice: Slicing Wireless Edge Computing Network with Decentralized Deep Reinforcement Learning. 2020.

LU, Y. et al. An access selection mechanism in 5G network slicing. In: **Proceedings - 2020 IEEE International Conference on Smart Internet of Things, SmartIoT 2020**. [S.l.]: IEEE, p. 72–78, 2020. ISBN 9781728165141.

MALCHE, T.; MAHESHWARY, P. Internet of things (iot) for building smart home system. In: **2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)**. [S.l.: s.n.], p. 65–70, 2017.

MATHEW, S.; MORDESON, J.; MALIK, D. **Fuzzy Graph Theory**. Springer International Publishing, 2017. (Studies in Fuzziness and Soft Computing). ISBN 9783319714073. Disponível em: <https://books.google.com.br/books?id=P-JEDwAAQBAJ>.

MEI, G. et al. A survey of internet of things (iot) for geohazard prevention: Applications, technologies, and challenges. **IEEE Internet of Things Journal**, v. 7, n. 5, p. 4371–4386, 2020.

MEI, J.; WANG, X.; ZHENG, K. Intelligent network slicing for v2x services toward 5g. **IEEE Network**, v. 33, n. 6, p. 196–204, 2019.

MOLINARO, A. et al. 5G-V2X communications and networking for connected and autonomous vehicles. **Future Internet**, v. 12, n. 12, p. 10–12, 2020. ISSN 19995903.

MONTGOMERY, D. C.; RUNGER, G. C. **Applied Statistics and Probability for Engineers**. [S.l.]: Wiley, 2018. v. 7. ISBN 978-1-119-40036-3.

MORDESON, J.; MATHEW, S. **Advanced Topics in Fuzzy Graph Theory**. Springer International Publishing, 2018. (Studies in Fuzziness and Soft Computing). ISBN 9783030042158. Disponível em: <https://books.google.com.br/books?id=D3B_DwAAQBAJ>.

MORGADO, A. et al. A survey of 5G technologies: regulatory, standardization and industrial perspectives. **Digital Communications and Networks**, v. 4, n. 2, p. 87–97, 2018. ISSN 23528648.

MOURADIAN, C. et al. A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges. **IEEE Communications Surveys and Tutorials**, IEEE, v. 20, n. 1, p. 416–464, 2018. ISSN 1553877X.

NABABAN, L. et al. Evaluation of lecturer performance during on covid-19 in improving of learning quality trough moora method at uph medan. In: **2021 9th International Conference on Cyber and IT Service Management (CITSM)**. [S.l.: s.n.], p. 1–6, 2021.

NGMN. **Recommendations for NGMN KPIs and Requirements for 5G**. 2016. 1-19 p.

NGMN. **Next Generation Mobile Networks Alliance**. 2023. Available online: <https://www.ngmn.org/> (accessed on 11 07 2023).

NGUYEN, H. Q. et al. A Comparative Study on Multi-Criteria Decision-Making in Dressing Process for Internal Grinding. **Machines**, v. 10, n. 5, 2022. ISSN 20751702.

NGUYEN, H. Q. et al. Multi-Criteria Decision Making in the PMEDM Process by Using MARCOS, TOPSIS, and MAIRCA Methods. **Applied Sciences (Switzerland)**, v. 12, n. 8, 2022. ISSN 20763417.

NS-3. **NS-3 Network Simulator**. 2022. Available online: <https://www.nsnam.org/> (accessed on 20 05 2022).

OECD. The Road to 5G Networks: Experience to Date and Future Developments. **OECD Digital Economy Papers**, n. 284, 2019.

ONF. **Open Networking Foundation**. 2023. Available online: <https://www.opennetworking.org> (accessed on 11 07 2023).

OPENAIRINTERFACE. **Open Air Interface**. 2022. Available online: <https://openairinterface.org/> (accessed on 18 12 2022).

OPENSTACK. **OpenStack**. 2022. Available online: <https://www.openstack.org/> (accessed on 21 12 2022).

ORDONEZ-LUCENA, J. et al. The Creation Phase in Network Slicing: From a Service Order to an Operative Network Slice. **2018 European Conference on Networks and Communications, EuCNC 2018**, p. 31–36, 2018.

OTOSHI, T. et al. Non-parametric Decision-Making by Bayesian Attractor Model for Dynamic Slice Selection. **2021 IEEE Global Communications Conference, GLOBECOM 2021 - Proceedings**, IEEE, 2021.

OUHIBI, A.; FRIKHA, H. Interval-valued intuitionistic fuzzy codas-sort method: Evaluation of natural resources in tunisia. In: **2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)**. [S.l.: s.n.], p. 1–5, 2020.

PAVANI, K.; SRIRAMYA, P. Enhancing public key cryptography using rsa, rsa-crt and n-prime rsa with multiple keys. In: **2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)**. [S.l.: s.n.], p. 1–6, 2021.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PEZZOTTI, N. et al. Approximated and user steerable tsne for progressive visual analytics. **IEEE Transactions on Visualization and Computer Graphics**, v. 23, n. 7, p. 1739–1752, 2017.

QIN, J. et al. General paradigm of edge-based internet of things data mining for geohazard prevention. **Big Data**, v. 9, n. 5, p. 373–389, 2021. PMID: 34227850. Disponível em: <https://doi.org/10.1089/big.2020.0392>.

R, N. D.; MOHANA. Jenkins pipelines: A novel approach to machine learning operations (mlops). In: **2022 International Conference on Edge Computing and Applications (ICECAA)**. [S.l.: s.n.], p. 1292–1297, 2022.

RAJ, E. et al. Edge mlops: An automation framework for aiot applications. In: **2021 IEEE International Conference on Cloud Engineering (IC2E)**. [S.l.: s.n.], p. 191–200, 2021.

RICART-SANCHEZ, R. et al. Towards an FPGA-Accelerated programmable data path for edge-to-core communications in 5G networks. **Journal of Network and Computer Applications**, Elsevier Ltd, v. 124, n. September, p. 80–93, 2018. ISSN 10958592. Disponível em: <https://doi.org/10.1016/j.jnca.2018.09.012>.

RIOS, V.; MONTEIRO, C.; GONDIM, P. Use of fuzzy logic for networks selection in heterogeneous wireless environment. 14th International Conference on Advanced Communication Technology (ICACT), v. 14th Inter, 2012.

RIVERA, J. J. D. et al. Network Slice Selection Function for Data Plane Slicing in a Mobile Network. In: **20th Asia-Pacific Network Operations and Management Symposium: Management in a Cyber-Physical World, APNOMS 2019**. [S.l.]: IEICE, p. 18–21, 2019. ISBN 9784885523205.

SAADON, G.; HADDAD, Y.; SIMONI, N. A survey of application orchestration and OSS in next-generation network management. **Computer Standards and Interfaces**, Elsevier, v. 62, n. November 2017, p. 17–31, 2019. ISSN 09205489. Disponível em: <https://doi.org/10.1016/j.csi.2018.07.003>.

SAHIDA, A. P.; SURARSO, B.; GERNOWO, R. The combination of the moora method and the copeland score method as a group decision support system (gdss) vendor selection. In: **2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)**. [S.l.: s.n.], p. 340–345, 2019.

SALAMA, A.; SAATCHI, R. Probabilistic classification of quality of service in wireless computer networks. **ICT Express**, Elsevier B.V., v. 5, n. 3, p. 155–162, 2019. ISSN 24059595. Disponível em: <https://doi.org/10.1016/j.icte.2018.09.001>.

SANTOS, J. F. et al. Breaking Down Network Slicing: Hierarchical Orchestration of End-to-End Networks. **IEEE Communications Magazine**, v. 58, n. 10, p. 16–22, 2020.

SHI, Y.; SAGDUYU, Y. E.; ERPEK, T. Reinforcement Learning for Dynamic Resource Optimization in 5G Radio Access Network Slicing. 2020.

SHI, Y. et al. How to Attack and Defend 5G Radio Access Network Slicing with Reinforcement Learning. **arXiv**, 2021.

SHURMAN, M.; RAWASHDEH, J.; JARADAT, A. Slice Selection in 5G Networks: Novel Approach for Accessing Multiple Slices Simultaneously. In: **2020 11th International Conference on Information and Communication Systems, ICICS**. [S.l.: s.n.], p. 113–117, 2020. ISBN 9781728162270.

SILVA, D. C. da et al. 5G Network Slice Selector in IoT Services Scenarios with QoS Requirements Guarantee. IEEE, p. 90–95, 2022. ISSN 15253511.

SILVA, D. C. da et al. A novel approach to multi-provider network slice selector for 5g and future communication systems. **Sensors**, v. 22, n. 16, 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/16/6066>.

SILVA, R. D. et al. Deployment options of 5g network slicing for smart healthcare. In: **2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)**. [S.l.: s.n.], p. 749–750, 2022.

SPARK. **Apache Spark**. 2022. <https://spark.apache.org/>. Available online; accessed on 19 12 2022.

SUNGKONO, K. R. et al. Optimizing time and cost activity based on discrete event simulation with multi-objective optimization method by ratio analysis (moora). In: **2022 International Conference on Advanced Computer Science and Information Systems (ICACSIS)**. [S.l.: s.n.], p. 53–58, 2022.

TALEB, T. et al. On Multi-Domain Network Slicing Orchestration Architecture and Federated Resource Control. **IEEE Network**, IEEE, v. 33, n. 5, p. 242–252, 2019. ISSN 1558-156X.

TEAGUE, K.; ABDEL-RAHMAN, M. J.; MACKENZIE, A. B. Joint Base Station Selection and Adaptive Slicing in Virtualized Wireless Networks: A Stochastic Optimization Framework. **2019 International Conference on Computing, Networking and Communications, ICNC 2019**, n. September, p. 859–863, 2019.

TOSCANO, M. et al. Machine Learning Aided Network Slicing. In: **2019 21st International Conference on Transparent Optical Networks (ICTON)**. [S.l.: s.n.], p. 1–4, 2019.

ULUTAş, A. et al. A new hybrid mcdm model for personnel selection based on a novel grey piprecia and grey OCRA methods. **Mathematics**, v. 8, n. 10, p. 1–14, 2020. ISSN 22277390.

USAMA, M. et al. Examining Machine Learning for 5G and Beyond through an Adversarial Lens. 2020.

VASILAKOS, X. et al. Integrated Methodology to Cognitive Network Slice Management in Virtualized 5G Networks. 2020.

VINCENZI, M.; LOPEZ-AGUILERA, E.; GARCIA-VILLEGAS, E. Maximizing Infrastructure Providers' Revenue through Network Slicing in 5G. **IEEE Access**, IEEE, v. 7, p. 128283–128297, 2019. ISSN 21693536.

WANG, J. et al. Performance model for video service in 5G networks. **Future Internet**, v. 12, n. 6, 2020. ISSN 19995903.

WANG, Z.; PEI, Y.; LI, J. A survey on search strategy of evolutionary multi-objective optimization algorithms. **Applied Sciences**, v. 13, n. 7, 2023. ISSN 2076-3417. Disponível em: <https://www.mdpi.com/2076-3417/13/7/4643>.

WĄTRÓBSKI, J. et al. Generalised framework for multi-criteria method selection. **Omega (United Kingdom)**, v. 86, p. 107–124, 2019. ISSN 03050483.

WEI, H.; ZHANG, Z.; FAN, B. Network slice access selection scheme in 5G. In: **Proceedings of the 2017 IEEE 2^{nd} Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2017**. [S.l.]: IEEE, p. 352–356, 2018. v. 1. ISBN 9781509064137.

WEN, R. et al. On Robustness of Network Slicing for Next-Generation Mobile Networks. **IEEE Transactions on Communications**, v. 67, n. 1, p. 430–444, 2019.

WIJETHILAKA, S.; LIYANAGE, M. Survey on network slicing for internet of things realization in 5g networks. **IEEE Communications Surveys Tutorials**, v. 23, n. 2, p. 957–994, 2021.

WU, Y.-J. et al. Network slicing for mmtc and urllc using software-defined networking with p4 switches. **Electronics**, v. 11, n. 14, 2022. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/11/14/2111>.

YACOUBI, S. et al. A multi-objective chaos game optimization algorithm based on decomposition and random learning mechanisms for numerical optimization. **Applied Soft Computing**, v. 144, p. 110525, 2023. ISSN 1568-4946. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1568494623005434>.

YAN, C.; SHENG, S. Sdn+k8s routing optimization strategy in 5g cloud edge collaboration scenario. **IEEE Access**, v. 11, p. 8397–8406, 2023.

YAZDANI, M. et al. A combined compromise solution (CoCoSo) method for multi-criteria decision-making problems. **Management Decision**, v. 57, n. 9, p. 2501–2519, 2019. ISSN 00251747.

YI, B. et al. A comprehensive survey of Network Function Virtualization. **Computer Networks**, v. 133, p. 212–262, 2018. ISSN 13891286.

YOU, X. et al. Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts. **Science China Information Sciences**, v. 64, n. 1, 2021. ISSN 18691919.

YOU, X. et al. AI for 5G: research directions and paradigms. **Science China Information Sciences**, v. 62, n. 2, p. 1–13, 2019. ISSN 18691919.

ZHANG, Y. et al. Flexible and anonymous network slicing selection for C-RAN enabled 5G service authentication. **Computer Communications**, Elsevier B.V., v. 166, n. December 2020, p. 165–173, 2021. ISSN 1873703X.

ZHANG, Z. et al. Dynamic slicing for deep neural networks. In: **2020 Proceedings of the 28th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering**. [S.l.: s.n.], p. 838–850, 2020. ISBN 9781450370431.

ZHAO, G. et al. Network slice selection in softwarization-based mobile networks. **Transactions on Emerging Telecommunications Technologies**, v. 31, n. 1, 2020. ISSN 21613915.

ÇATAK, E.; DURAK-ATA, L. Waveform design considerations for 5g wireless networks. In: BIZAKI, H. K. (Ed.). **Towards 5G Wireless Networks**. Rijeka: IntechOpen, 2016. cap. 2. Disponível em: <https://doi.org/10.5772/66050>.

APPENDIX A

Below is a didactic example of the execution of the Algorithm 1, regarding the use of Promethee II and TOPSIS. In addition, the VIKOR and COPRA methods were described for the slice selection function.

A.1 Details of MCDM Methods

A.1.1 Application of the Promethee II Method

The Promethee II method considered four slices as alternatives, according to the organization shown in Table 13. In this example, the criteria values for each of the alternatives have been set arbitrarily. Next, the type of curve to be used in the comparison between the alternatives was defined, for each of the considered criteria (ALINEZHAD; KHALILI, 2019; FIGUEIRA; GRECO; EHRGOTT, 2005).

Table 14 presents this information in a consolidated form. It is important to note that the function curves, also known as preference functions, follow the equation (A.1) for the curve I (usual criterion). For such a curve, each alternative is compared with the other, not depending on additional parameters. The alternative that presents the best value for a given criterion is the winner in light of this criterion. This process is called strict immediate preference.

Table 13 – Criteria and Alternatives Matrix - Promethee II.

Alternatives	Latency	Jitter	Loss	Reliability
Slice 1	100.85	89.10	20	54
Slice 2	85.65	34.65	18	62
Slice 3	92.40	66.20	24	31
Slice 4	76.80	48.30	17	50

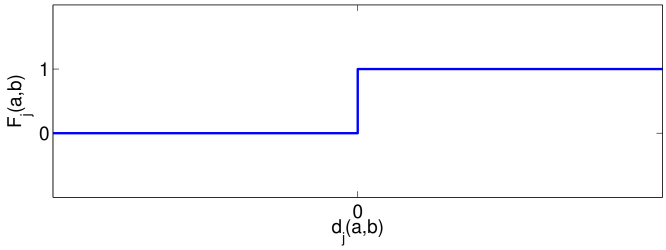
Source: elaborated by the author.

Table 14 – Characteristics of the criteria for the decision maker.

Criteria	Weight	Curve	Parameters
Latency	0.27	I	—
Jitter	0.19	I	—
Loss	0.47	I	—
Reliability	0.07	I and V	p = *60% and q=45%*

Source: elaborated by the author.

Figure 41 – Promethee II - Curve I: Usual Criterion.



For the curve V (indifference criterion), which uses the equation (A.2), indifference (q) and preference (p) thresholds are defined. The preference among the alternatives is growing for this function (FIGUEIRA; GRECO; EHRGOTT, 2005). Fig. 41 and 42 presents the curves described in the equations (A.1) and (A.2).

$$\begin{aligned} d_j(a,b) \leq 0, F_j(a,b) = 0 \\ d_j(a,b) > 0, F_j(a,b) = 1 \end{aligned} \tag{A.1}$$

$$\begin{aligned} d_j(a,b) \leq q, F_j(a,b) = 0 \\ q < d_j(a,b) \leq p, F_j(a,b) = \tfrac{d-q}{p-q} \\ d_j(a,b) > p, F_j(a,b) = 1 \end{aligned} \tag{A.2}$$

In the equations (A.1) and (A.2), $d_j(a,b)$ refers to the matrix with the alternatives, where the alternative **a** is represented vertically and **b** horizontally. Comparisons between alternatives occur according to a certain criterion, as illustrated in Table 15. $F_j(a,b)$ represents the preference function, which assumes values between 0 and 1, indicating the

Figure 42 – Promethee II - Curve V: Indifference Criterion.

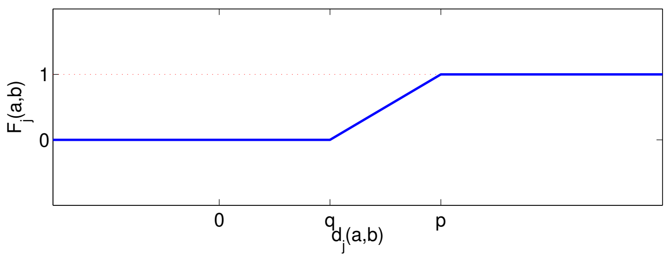


Table 15 – Comparison matrix in the light of the latency criterion.

Latency	Slice 1	Slice 2	Slice 3	Slice 4
Slice 1	—	0	0	0
Slice 2	1	—	1	0
Slice 3	1	0	—	0
Slice 4	1	1	1	—

Source: elaborated by the author.

preference between the alternatives. For the equation (A.2) we also have the limits of indifference (q) and preference (p).

The degree of over-classification $\pi(a,b)$, calculated by comparing the alternatives **a** and **b**, is given by the set of equations (A.3), where w_j represents the criterion weight j, $F_j(a,b)$ the preference function, g_j refers to the criterion value considering the curve and the weight used, $\boldsymbol{A}$ the matrix for comparing the alternatives in relation to a given criterion, and **a** and **b**, as previously stated, are two arbitrary alternatives. This scheme is detailed below, according to the comparison matrices for each of the criteria (see Tables 15, 16, 17 and 18).

$$\begin{gathered}\pi(\mathbf{a},\mathbf{b}) = \tfrac{1}{W}\sum_{j=1}^{n} w_j \cdot F_j(\mathbf{a},\mathbf{b}),\ \text{where } W = n \\ d_j : \mathbf{A}\times\mathbf{A} \rightarrow R \\ d_j(\mathbf{a},\mathbf{b}) = g_j(\mathbf{a}) - g_j(\mathbf{b}),\ \text{where } \mathbf{a},\ \mathbf{b}\ \in \mathbf{A}. \\ F_j(\mathbf{a},\mathbf{b}) = F_j[d_j(\mathbf{a},\mathbf{b})]\end{gathered} \tag{A.3}$$

The definition of the comparison matrices by criteria follows the format given by Table 13. It is important to point out that for the latency, jitter and packet loss criteria, the smaller the values obtained, the better the slice is in relation to the parameter in question. Based on this premise, it was necessary to invert the (A.1) equations, so that the preference functions reflected this premise. The change is demonstrated in the equation (A.4).

$$\begin{gathered} d_j(a,b) \geq 0, F_j(a,b) = 0 \\ d_j(a,b) < 0, F_j(a,b) = 1 \end{gathered} \tag{A.4}$$

After defining the comparison matrices for each criterion, the weights provided by the K-Means algorithm are applied, in this example, described in Table 14. These operations can be viewed in the Tables 19, 20, 21 and 22.

Then, the so-called positive (ϕ^{+}) and negative (ϕ^{-}) preference indices are calculated, according to the equation (A.5). The first corresponds to the preference of alternative

Table 16 – Comparison matrix in the light of the jitter criterion.

Jitter	Slice 1	Slice 2	Slice 3	Slice 4
Slice 1	—	0	0	0
Slice 2	1	—	1	1
Slice 3	1	0	—	0
Slice 4	1	0	1	—

Source: elaborated by the author.

Table 17 – Comparison matrix in the light of the loss criterion.

Loss	Slice 1	Slice 2	Slice 3	Slice 4
Slice 1	—	0	1	0
Slice 2	1	—	1	0
Slice 3	0	0	—	0
Slice 4	1	1	1	—

Source: elaborated by the author.

Table 18 – Comparison matrix in the light of the reliability criterion.

Loss	Slice 1	Slice 2	Slice 3	Slice 4
Slice 1	—	0	1	1
Slice 2	1	—	1	1
Slice 3	0	0	—	0
Slice 4	0	0	1	—

Source: elaborated by the author.

Table 19 – Latency comparison matrix considering the criterion weight.

Latency	Slice 1	Slice 2	Slice 3	Slice 4
Slice 1	—	$0 \times 0.27 = 0$	$0 \times 0.27 = 0$	$0 \times 0.27 = 0$
Slice 2	$1 \times 0.27 = 0.27$	—	$1 \times 0.27 = 0.27$	$0 \times 0.27 = 0$
Slice 3	$1 \times 0.27 = 0.27$	$0 \times 0.27 = 0$	—	$0 \times 0.27 = 0$
Slice 4	$1 \times 0.27 = 0.27$	$1 \times 0.27 = 0.27$	$1 \times 0.27 = 0.27$	—

Source: elaborated by the author.

Table 20 – Jitter comparison matrix considering the criterion weight.

Jitter	Slice 1	Slice 2	Slice 3	Slice 4
Slice 1	—	$0 \times 0.19 = 0$	$0 \times 0.19 = 0$	$0 \times 0.19 = 0$
Slice 2	$1 \times 0.19 = 0.19$	—	$1 \times 0.19 = 0.19$	$1 \times 0.19 = 0.19$
Slice 3	$1 \times 0.19 = 0.19$	$0 \times 0.19 = 0$	—	$0 \times 0.19 = 0$
Slice 4	$1 \times 0.19 = 0.19$	$0 \times 0.19 = 0$	$1 \times 0.19 = 0.19$	—

Source: elaborated by the author.

Table 21 – Loss comparison matrix considering the criterion weight.

Loss	Slice 1	Slice 2	Slice 3	Slice 4
Slice 1	—	$0 \times 0.47 = 0$	$1 \times 0.47 = 0.47$	$0 \times 0.47 = 0$
Slice 2	$1 \times 0.47 = 0.47$	—	$1 \times 0.47 = 0.47$	$0 \times 0.47 = 0$
Slice 3	$0 \times 0.47 = 0$	$0 \times 0.47 = 0$	—	$0 \times 0.47 = 0$
Slice 4	$1 \times 0.47 = 0.47$	$1 \times 0.47 = 0.47$	$1 \times 0.47 = 0.47$	—

Source: elaborated by the author.

Table 22 – Reliability comparison matrix considering the criterion weight.

Reliability	Slice 1	Slice 2	Slice 3	Slice 4
Slice 1	—	$0 \times 0.07 = 0$	$1 \times 0.07 = 0.07$	$1 \times 0.07 = 0.07$
Slice 2	$1 \times 0.07 = 0.07$	—	$1 \times 0.07 = 0.07$	$1 \times 0.07 = 0.07$
Slice 3	$0 \times 0.07 = 0$	$0 \times 0.07 = 0$	—	$0 \times 0.07 = 0$
Slice 4	$0 \times 0.07 = 0$	$0 \times 0.07 = 0$	$1 \times 0.07 = 0.07$	—

Source: elaborated by the author.

a over all alternatives, while the negative preference index corresponds to the preference index of all alternatives over **a**.

$$\begin{aligned} \phi^+(a) &= \textstyle\sum_{b \in A} \pi(a,b) \\ \phi^-(a) &= \textstyle\sum_{b \in A} \pi(b,a) \end{aligned} \tag{A.5}$$

The positive and negative preference matrix is presented in Table 23.

Finally, the matrix resulting from the application of the method is calculated, that is, the final index of preference, given by the equation (A.6). The sum of the rows in Table 23 corresponds to the positive preference index (ϕ^+), and the sum of the columns to the negative preference index (ϕ^-).

Table 23 – Positive and negative preference matrix.

π	Slice 1	Slice 2	Slice 3	Slice 4
Slice 1	—	0	0.54	0.27
Slice 2	1	—	1	0.26
Slice 3	0.46	0	—	0
Slice 4	0.93	0.74	1	—

Source: elaborated by the author.

Table 24 – Promethee II Resultant Matrix.

Alternatives	ϕ^+	ϕ^-	ϕ
Slice 1	0.81	2.39	-1.58
Slice 2	2.26	0.74	1.52
Slice 3	0.46	2.54	-2.08
Slice 4	2.67	0.53	2.14

Source: elaborated by the author.

$$\begin{gathered}\phi(a) = \phi^+(a) - \phi^-(a), a \in A. \\ \phi(a) > \phi(b), \text{then alternative } \mathbf{a} \text{ is preferred over alternative } \mathbf{b}. \\ \phi(a) = \phi(b), \text{then alternative } \mathbf{a} \text{ is indifferent to alternative } \mathbf{b}.\end{gathered} \tag{A.6}$$

The preference ranking between the alternatives (ϕ), output of the Promethee II method, is shown in Table 24.

The sort order for this example is therefore: ***Slice 4*** $>$ ***Slice 2*** $>$ ***Slice 1*** $>$ ***Slice 3***. Therefore, ***Slice 4*** is the slice selected as the best among those evaluated.

A.1.2 Application of the TOPSIS Method

The TOPSIS method employs the principle of choosing an alternative that is closest to the positive ideal solution (best solution), and farthest from the negative ideal solution (worst solution). Thus, the method focuses on maximizing benefits and minimizing costs (BAKMAZ; BOJKOVIC; BAKMAZ, 2020).

The execution of the method consists of 06 steps, which are demonstrated below. To simplify the demonstration, only 03 criteria will be considered and 02 alternatives (slices). Table 25 presents the initial matrix.

Table 25 – Criteria values for Slice 1 and Slice 2.

Alternatives	Latency	Jitter	Loss
Slice 1	0.85	0.100	1
Slice 2	0.65	0.65	5

Source: elaborated by the author.

Step 01: normalize the data as in Table 26, according to the equation (A.7).

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}} \qquad for\ i = 1, ..., m;\ and\ j = 1, ..., n \tag{A.7}$$

Table 26 – Normalization of Slice 1 and Slice 2 criteria.

Alternatives	Latency	Jitter	Loss
Slice 1	$\frac{0.85}{\sqrt{(0.85)^2+(0.100)^2+(1)^2}}$	$\frac{0.100}{\sqrt{(0.85)^2+(0.100)^2+(1)^2}}$	$\frac{1}{\sqrt{(0.85)^2+(0.100)^2+(1)^2}}$
Slice 2	$\frac{0.65}{\sqrt{(0.65)^2+(0.65)^2+(5)^2}}$	$\frac{0.65}{\sqrt{(0.65)^2+(0.65)^2+(5)^2}}$	$\frac{5}{\sqrt{(0.65)^2+(0.65)^2+(5)^2}}$

Source: elaborated by the author.

Table 27 – Weighted normalization of the Slice 1 and Slice 2.

Alternatives	Latency	Jitter	Loss
Slice 1	0.64*0.24 = **0.15**	0.07*0.14 = **0.01**	0.75*0.62 = **0.47**
Slice 2	0.12*0.24 = **0.02**	0.12*0.14 = **0.01**	0.96*0.62 = **0.59**

Source: elaborated by the author.

Where x_{ij} are the matrix values containing the alternatives in each line, by the criteria in each column.

Step 02: calculate the weighted normalization, that is, multiply the weights of each criterion by the normalized data of the matrix r_{ij} according to Table 27, according to equation (A.8).

$$v_{ij} = w_j * r_{ij} \qquad for\ i = 1, ..., m;\ and\ j = 1, ..., n \tag{A.8}$$

Being w_j a vector containing the weights provided by the K-means algorithm.

Step 03: identify the ideal positive solutions and ideal negative solutions according to Table 28, using the equations (A.9) and (A.10).

$$A^+ = \{v_1^+, ..., v_j^+, ..., v_n^+\} = \{(max_j v_{ij} \mid j = 1, ..., n) \mid i = 1, ..., m\} \tag{A.9}$$

$$A^- = \{v_1^-, ..., v_j^-, ..., v_n^-\} = \{(min_j v_{ij} \mid j = 1, ..., n) \mid i = 1, ..., m\} \tag{A.10}$$

The equation (A.9) seeks to select the highest values of the criteria, that is, the higher

Table 28 – Positive and negative ideal solutions of Slices 1 and 2.

Alternatives	Latency	Jitter	Loss
S+	0.02	0.01	0.47
S-	0.15	0.01	0.59

Source: elaborated by the author.

Table 29 – Solutions of the ideal positive and negative distances of Slices 1 and 2.

Slices		Positive ideal and negative ideal distance	Total
Slice 1	D+	$\sqrt{((0.15-0.02)^2)+((0.01-0.01)^2)+((0.47-0.47)^2)}$	**0.169**
Slice 1	D-	$\sqrt{((0.15-0.15)^2)+((0.01-0.01)^2)+((0.47-0.59)^2)}$	**0.144**
Slice 2	D+	$\sqrt{((0.02-0.02)^2)+((0.01-0.01)^2)+((0.59-0.47)^2)}$	**0.144**
Slice 2	D-	$\sqrt{((0.02-0.15)^2)+((0.01-0.01)^2)+((0.59-0.59)^2)}$	**0.169**

Source: elaborated by the author.

Table 30 – Approximation with the ideal positive and negative solutions.

Slices	Result
Slice 1	$\frac{0.144}{0.169+0.144} = \mathbf{0.46}$
Slice 2	$\frac{0.169}{0.144+0.169} = \mathbf{0.54}$

Source: elaborated by the author.

the better. In this way, criteria such as bandwidth, reliability, for example, are selected. The equation (A.10) selects the smallest values, in this case, the smaller the better. Thus, criteria such as: jitter, latency and loss are selected (BAKMAZ; BOJKOVIC; BAKMAZ, 2020).

Step 04: find the ideal positive and ideal negative distances for each alternative, according to Table 29, from the equations (A.11) and (A.12).

$$D_i^+ = \sqrt{\sum_{j=1}^{N}(v_{ij} - s_j^+)^2} \qquad for\ i = 1, ..., N \tag{A.11}$$

$$D_i^- = \sqrt{\sum_{j=1}^{N}(v_{ij} - s_j^-)^2} \qquad for\ i = 1, ..., N \tag{A.12}$$

Where the values selected in step 03 are placed in s_j^+ and s_j^- and the matrix values in v_{ij}. Obtaining, therefore, a vector with the largest (D_i^+) and smallest (D_i^-) distances of each alternative.

Step 05: calculate the relative approximation with the positive ideal solution and the negative ideal solution, according to Table 30, and using the equation (A.13).

$$A_i = \frac{D_i^-}{D_i^+ + D_i^-} \qquad for\ 0 \leq A_i \leq 1,\ i = 1, 2, 3, ...\ M \tag{A.13}$$

Step 06: the slices are ordered according to the approximation of the ideal solution. In this case, the slice that has the closest score to 1 and the worst case distance (0), is selected. For this example, **Slice 2** is selected, finalizing the process.

A.1.3 Description of the VIKOR Method

According to (ALINEZHAD; KHALILI, 2019; WĄTRÓBSKI et al., 2019; BABASHAMSI et al., 2016), the following steps make up the VIKOR method:

Step 01: Consider Saaty's fundamental scale (FIGUEIRA; GRECO; EHRGOTT, 2005). Table 31 presents the relationship between the intensity of importance (preference) of the criteria and the verbal definition of the same, i.e., the relationship between the numerical and verbal scale. Intermediate values (2,4,6,8) attribute definitions immediately below the scale of importance.

Also consider the Table 32, containing the IR (Random Index) provided by Thomas L. Saaty (FIGUEIRA; GRECO; EHRGOTT, 2005).

Table 31 – Saaty's Fundamental Scale.

Intensity of importance	Definition
1	Equal preference
3	Moderate preference
5	Strong preference
7	Very strong preference
9	Absolute preference

Source: Adapted from Thomas L. Saaty (1980).

From Tables 31 and 32, and using the set of data defined in Table 13, determine the value f_j^* and the worst value f_j^- of each criterion, according to the equations (A.14) and (A.15).

$$f_j^* = max_i \quad f_{ij} \tag{A.14}$$

$$f_j^- = min_i \quad f_{ij} \tag{A.15}$$

Step 02: Calculate the values S_i and R_i, according to the equations (A.16) and

Table 32 – Random Index (RI).

Matrix size (n)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.48	1.56	1.57	1.59

Source: Thomas L. Saaty (1980).

(A.17).

$$S_i = \sum_{j=1}^{m} w_j \frac{f_j^* - f_{ij}}{f_j^* - f_i^-} \tag{A.16}$$

$$R_i = max_j \left[w_j \frac{f_j^* - f_{ij}}{f_j^* - f_i^-} \right] \tag{A.17}$$

Where w_j is the criterion weight.

Step 03: Determine the value of Q_i, where $i = 1, ..., n$, obtained by the equation (A.18).

$$Q_i = v \left[\frac{(S_i - S^*)}{(S^- - S^*)} \right] + (1 - v) \left[\frac{(R_i - R^*)}{(R^- - R^*)} \right] \tag{A.18}$$

Where $S^* = min_i S_i$, $S^- = max_i S_i$, $R^* = min_i R_i$, $R^- = max_i R_i$ and v used as maximum group utility. Default value: $v = 0.5$.

Step 04: Then sort S_i, R_i, Q_i in descending order, if the two conditions (A.19) and (A.20) are satisfied.

$$(i)\ Q(a_2) - Q(a_1) \geq (1/(n-1)) \tag{A.19}$$

$$(ii)\ Q(a_m) - Q(a_1) < (1/(n-1)) \tag{A.20}$$

Step 05: Sort the slices, finalizing the process.

A.1.4 Description of the COPRAS Method

The COPRAS method is simpler when compared to other decision-making methods, which facilitates its application in several areas.

According to (ALINEZHAD; KHALILI, 2019), the following steps make up the COPRAS method:

Step 01: Initially, the decision matrix must be considered. In this work represented by Table 13.

Step 02: The Normalization of Decision-Making Matrix.

In order to allow the comparison between the alternatives, use the normalization process given by the equation (A.21).

$$r_{ij}^{*} = \frac{r_{ij}}{\sum_{i=1}^{m} r_{ij}}; \quad j = 1, ..., n \tag{A.21}$$

Where, r_{ij}^{*} indicates the normalized value of the decision matrix of i th alternative in j th attribute.

Step 03: Determining of Weighted Normalized Decision-Making Matrix.

After the normalization process, the weighted matrix is obtained by multiplying the attribute weights.

The (A.22) equation is used.

$$r_{ij}^{-} = r_{ij}^{*}.w_j; \quad i = 1, ..., m; \quad j = 1, ..., n \tag{A.22}$$

Where w_j is the weight of attribute $[w_1, w_2, ..., w_n]$.

Step 04: Calculation of Maximizing and Minimizing Indexes

In this step, considering the type of each attribute (negative or positive), the maximizing and minimizing indexes of each attribute are obtained.

Equations (A.23) and (A.24) demonstrate this process.

$$S_{+i} = \sum_{j=1}^{g} r_{ij}^{-}; \quad i = 1, ..., m \tag{A.23}$$

$$S_{-i} = \sum_{j=g+1}^{g} r_{ij}^{-}; \quad i = 1, ..., m \tag{A.24}$$

Where g is the number of positive attributes and $n - g$ is the number of negative attributes, and S_i defines the maximizing and minimizing indexes of i th attribute.

Step 05: Calculation of the Relative Significance Value

The calculation of the relative significance value (Q_i) for each alternative is obtained through the equations (A.25) and (A.26).

$$Q_i = S_{+i} + \frac{min_i S_{-i} \sum_{i=1}^{m} S_{-i}}{S_{-i} \sum_{i=1}^{m} \frac{min_i S_{-i}}{S_{-i}}} \tag{A.25}$$

$$Q_i = S_{+i} + \frac{\sum_{i=1}^{m} S_{-i}}{S_{-i} \sum_{i=1}^{m} \frac{1}{S_{-i}}} \tag{A.26}$$

Step 06: The priority order of alternatives

The final ranking of alternatives considers the relative significance values in descending order. Thus, the highest final value has the highest rank, finalizing the process.

A.1.5 Other MCDM Methods

The other decision-making methods used in the NSSF DAF, especially in scenario 03 described in Section 5.2.3, use similar approaches, considering the particularities and characteristics of each one of them.

In fact, a set of routines is necessary to evaluate the data set obtained in various environments test, defining the decision matrix format. Following this, several matrix operations are performed to evaluate the alternatives, considering the different weights of the criteria established by the user or by another strategy. This behavior in this work is configured through utility functions defined by the proposed hybrid algorithms.

Thus, from the evaluation of the alternatives, and considering the weights and preferences of the user or system, the final ranking is obtained, and the best slice is selected in network runtime.

For more information about the methods, see the following references:(NGUYEN et al., 2022a; WANG et al., 2020; ULUTAş et al., 2020; DEZERT et al., 2020; KUNDAKCı, 2019; YAZDANI et al., 2019; ALINEZHAD; KHALILI, 2019; WĄTRÓBSKI et al., 2019; HEZER; GELMEZ; ÖZCEYLAN, 2021; BOJKOVIC; BAKMAZ; BAKMAZ, 2019; BABASHAMSI et al., 2016; FIGUEIRA; GRECO; EHRGOTT, 2005).

APPENDIX B

This appendix explains the experiments performed in the TS-01 (experiments 2,3 and 4) and T-03 (experiment 2 and priority data) scenarios. The experiments follow the same approach explained in Chapter 6, and their objectives is to verify the performance of the different approaches considering the weight setups presented in Tables 7 and 8.

Thus, it was decided to use a linear regression model by applying the corrections provided by the *sqrt* function. As a result, this strategy approximated the distribution to a normal one, allowing the use of an ANOVA model and applying multiple comparison methods.

B.1 TS-01: Experiment 2

Table 33 presents the methods results for each slices in Experiment 2. The average performance of the methods is summarized in Figure 43.

For Experiment 2, the attributes *"Latency", "Jitter", "Loss"* were prioritized, according to the weights defined in Table 8. The weights correspond to the result of the K-means algorithm analysis, discussed in Section 5.3.1. This configuration prioritizes Slice 1 and 2, with a slight advantage for Slice 2. In this sense, we seek to verify the MCDM methods sensitivity to these slices requirements. In addition, we aim to verify statistically whether there are significant differences in the accuracy of selection between the evaluated methods.

For Slices 1, 2 and 3, there was normality violation, independence of residuals and homoscedasticity of variances, which implied the rejection of the null hypotheses defined in equations (6.2), (6.3), and (6.4). Solving this situation, the KRUSKAL-WALLIS non-parametric method was used, where it was found that the median of the evaluated methods are different, and that Ocra and Cocoso methods present the lowest IQR, which corroborates with the descriptive analysis shown in Figure 44.

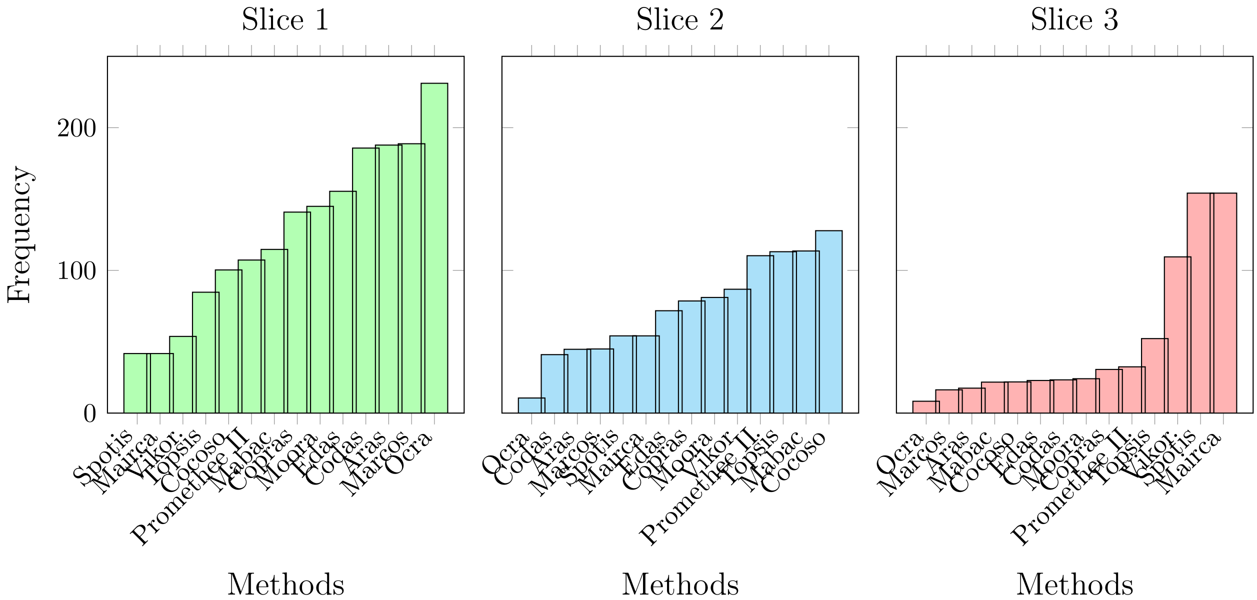
Table 33 – Methods Results: TS01 - Experiment 2

Slice	Methods	Mean	STD	Var	CI
Slice 1	Aras	187.82	21.81	475.59	180.09 – 195.55
	Cocoso	100.33	32.50	1056.23	88.81 – 111.86
	Codas	185.73	22.82	520.64	177.64 – 193.82
	Copras	140.85	55.14	3040.38	121.30 – 160.40
	Edas	155.42	55.16	3042.44	135.87 – 174.98
	Mabac	114.70	60.14	3616.97	93.37 – 136.02
	Mairca	41.73	26.03	677.33	32.50 – 50.96
	Marcos	188.76	24.19	585.06	180.18 – 197.33
	Moora	144.88	59.42	3530.80	123.81 – 165.95
	Ocra	231.12	4.61	21.30	229.48 – 232.76
	Promethee II	107.27	50.79	2579.83	89.26 – 125.28
	Spotis	41.73	26.03	677.33	32.50 – 50.96
	Topsis	84.70	51.88	2691.47	66.30 – 103.09
	Vikor	53.73	28.63	819.70	43.58 – 63.88
Slice 2	Aras	44.70	30.77	946.84	33.79 – 55.61
	Cocoso	127.82	37.28	1389.84	114.60 – 141.04
	Codas	40.97	35.86	1286.28	28.25 – 53.69
	Copras	78.58	62.25	3875.56	56.50 – 100.65
	Edas	71.73	58.49	3421.58	50.99 – 92.47
	Mabac	113.61	73.26	5367.18	87.63 – 139.58
	Mairca	54.12	57.47	3303.11	33.74 – 74.50
	Marcos	44.94	33.09	1094.81	33.21 – 56.67
	Moora	81.03	64.39	4146.47	58.20 – 103.86
	Ocra	10.58	6.96	48.50	8.11 – 13.05
	Promethee II	110.30	68.29	4663.03	86.09 – 134.52
	Spotis	54.12	57.47	3303.11	33.74 – 74.50
	Topsis	113.09	88.99	7918.46	81.54 – 144.64
	Vikor	86.79	80.53	6485.86	58.23 – 115.34
Slice 3	Aras	17.48	15.01	225.38	12.16 – 22.81
	Cocoso	21.85	7.59	57.63	19.16 – 24.54
	Codas	23.30	18.37	337.47	16.79 – 29.82
	Copras	30.58	22.56	508.81	22.58 – 38.57
	Edas	22.85	17.00	288.95	16.82 – 28.88
	Mabac	21.70	25.85	668.34	12.53 – 30.86
	Mairca	154.15	50.50	2550.63	136.24 – 172.06
	Marcos	16.30	14.86	220.78	11.03 – 21.57
	Moora	24.09	19.80	391.90	17.07 – 31.11
	Ocra	8.30	3.36	11.28	7.11 – 9.49
	Promethee II	32.42	25.16	633.06	23.50 – 41.35
	Spotis	154.15	50.50	2550.63	136.24 – 172.06
	Topsis	52.21	48.84	2385.36	34.89 – 69.53
	Vikor	109.48	65.99	4354.57	86.09 – 132.88

Legend: STD: Standard Deviation; Var: Variance ; CI: Confidence Intervals.

Source: elaborated by the author.

Figure 43 – Behavior of the NSSF DAF methods - TS 01: Experiment 2



Source: elaborated by the author.

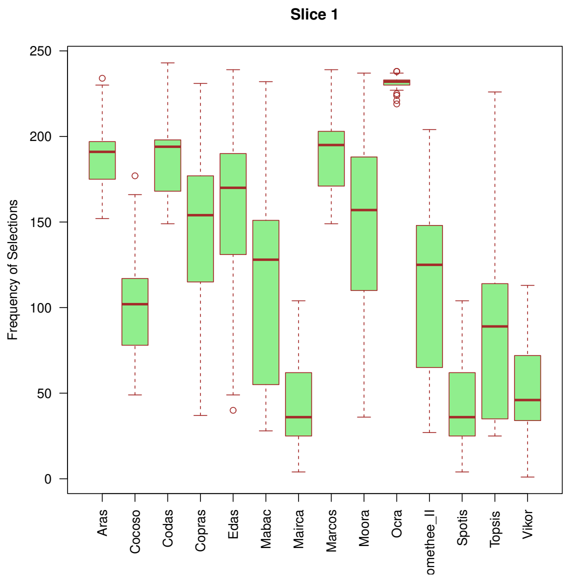
In addition, to allow the performance comparison of all methods for each of the slices, a linear regression model was employed, and the result was summarized in an ANOVA. This procedure was possible by applying the $sqrt()$ function to the response variable (Frequency of Selections), which made the distribution of this variable approach a normal distribution, as illustrated in Figures 46b, 47b, and 48b. Note that although the response variable has characteristics of count data, it does not obey a binomial distribution, as shown in the simulated envelope plots of Figures 46a, 47a, 48a, and extensively discussed in Section 6.2.1.

Finally, from the estimated coefficients analysis in the regression model, it appears that the Ocra method reached the highest positive number (1.52), which demonstrates its superiority in relation to the others for Slice 1. Regarding Slice 2, it was observed that the Cocoso (5.02), Mabac (3.85), Promethee II (3.79) and Topsis (3.42) methods demonstrate superiority in selection accuracy.

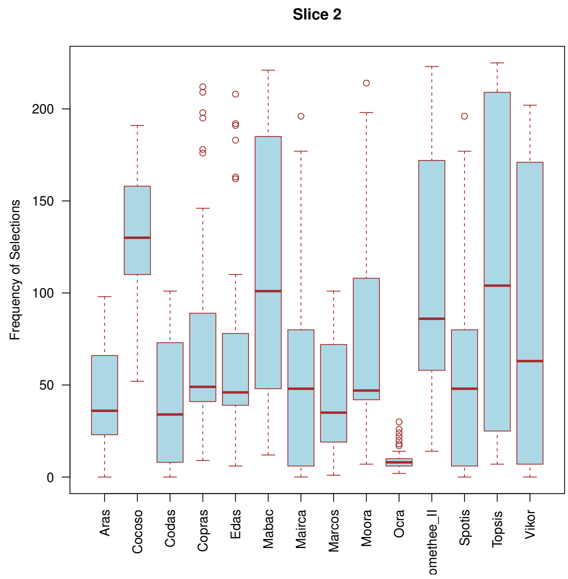
For Slice 3, it was observed that Spotis (8.57), Mairca (8.57) and Vikor (6.28) method performed better. However, this finding is dissonant from the other methods, since for Experiment 2 the attributes with the highest weight were Latency, Jitter and Loss, and considering the characteristics of the slices from the machine learning model used, the slices with the highest probability of being selected would be Slices 1 and 2.

The result of the multivariate comparison provided by the TukeyHSD test for Experiment 2, considering a significance level of 95%, is presented in Figure 45. For Slice 1, in 26 comparisons there is no significant difference between the methods, this is denoted by the presence of three clusters in Figure 45a. For Slice 2, in 16 comparisons there is no superiority between the methods, as evidenced in Figure 45b. Finally, for Slice 3, there is greater uniformity in the performance of the methods, with the exception of the Mairca and Spotis methods. However, in 50 out of 182 comparisons, there is no significant difference between the methods, as demonstrated by the cluster density in Figure 45c.

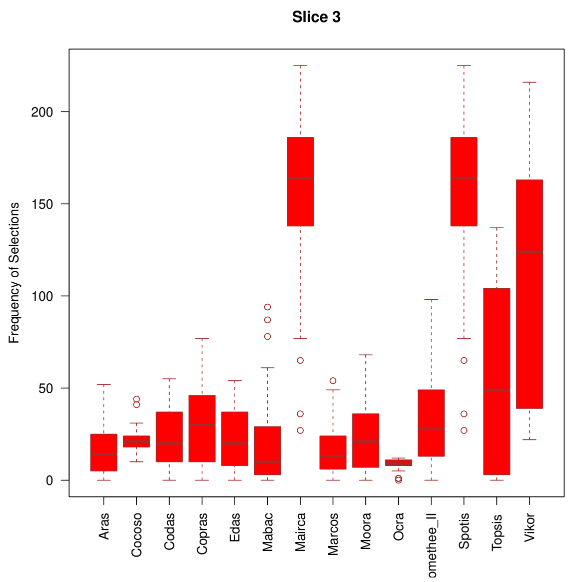
Figure 44 – Descriptive analysis using boxplot - TS 01: Experiment 2.



(a) Slice 1



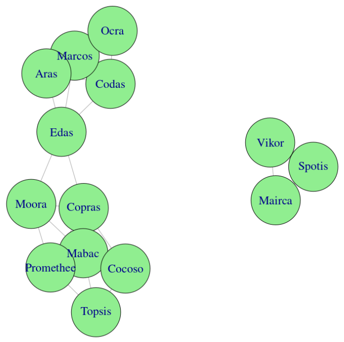
(b) Slice 2



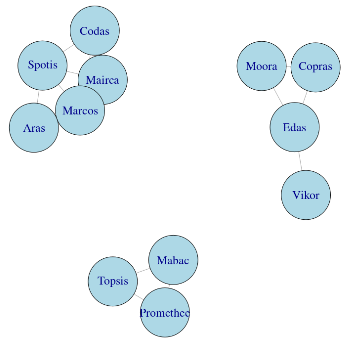
(c) Slice 3

Source: elaborated by the author.

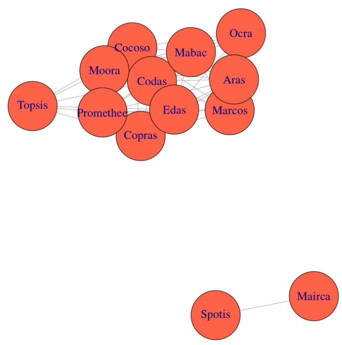
Figure 45 – Network graph TS01: Experiment 2. There is no significant difference between this methods.



(a) Slice 1



(b) Slice 2

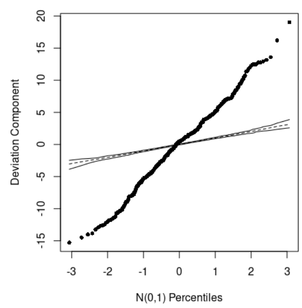


(c) Slice 3

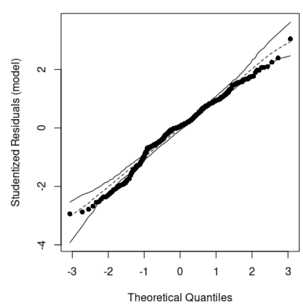
Legend: 95% confidence intervals comparing each pair of methods.

Source: elaborated by the author.

Figure 46 – Simulated envelope TS01: Experiment 2 - Slice 1.



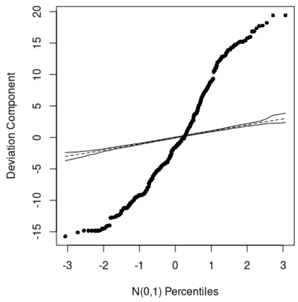
(a) Binomial GLM.



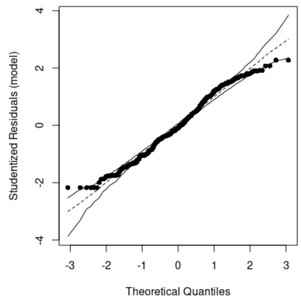
(b) Normal LM.

Source: elaborated by the author.

Figure 47 – Simulated envelope TS01: Experiment 2 - Slice 2.



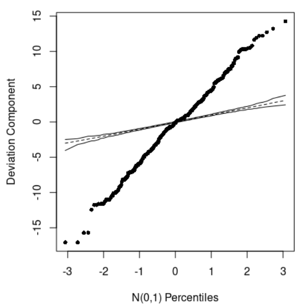
(a) Binomial GLM.



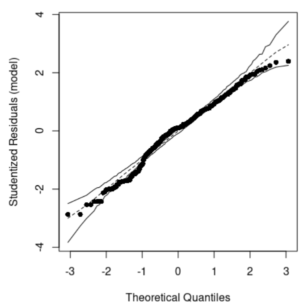
(b) Normal LM.

Source: elaborated by the author.

Figure 48 – Simulated envelope TS01: Experiment 2 - Slice 3.



(a) Binomial GLM.



(b) Normal LM.

Source: elaborated by the author.

B.2 TS-01: Experiment 3

For Experiment 3, the attributes *"Bandwidth", "Transfer", "Loss"* were prioritized, according to the weights defined in Table 8. This configuration prioritizes Slices 1 and 3, with a slight advantage for Slice 1. It is intended to verify whether there are significant differences in the selection accuracy between the evaluated methods, i.e., the validation of the main hypothesis defined in the equation (6.1).

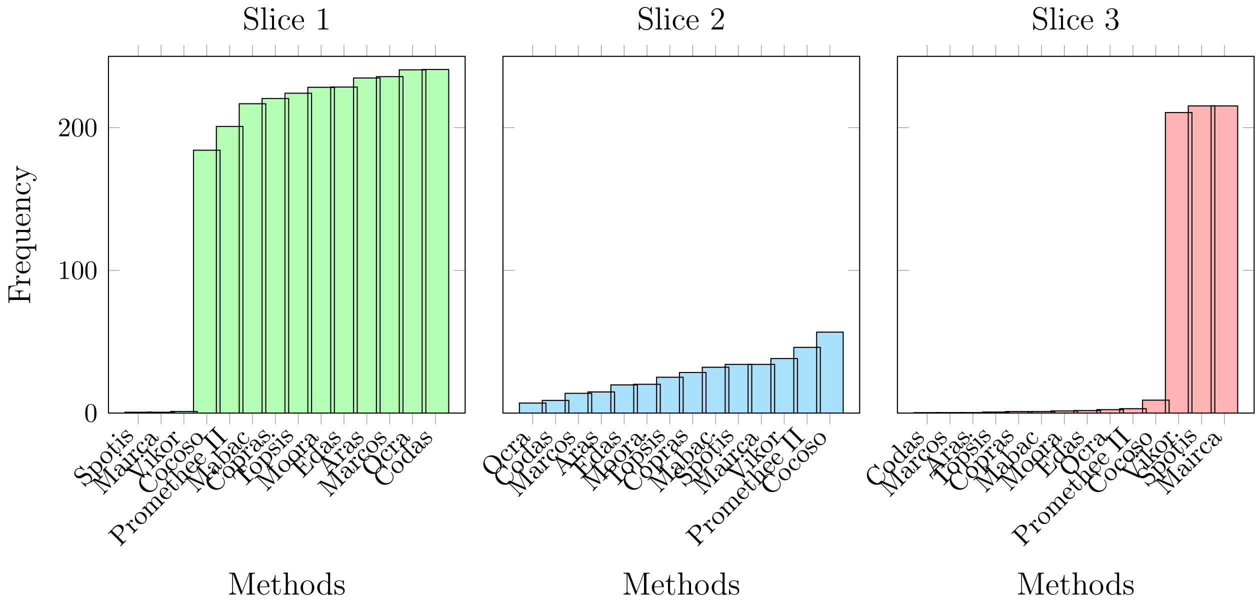
For Slice 1, and considering the result of the KRUSKAL-WALLIS non-parametric test, it was found that the medians of the evaluated methods are different, and that the Ocra method presents the lowest IQR, followed by the Codas, Aras and Marcos methods, which corroborates the descriptive analysis provided in Figure 50a. For the Mairca, Spotis and Vikor methods, the interquartile range was zero, which demonstrates the inefficiency of these methods for the selection of Slice 1.

From analyzing the estimated coefficients in the regression model, it appears that the Codas method reached the highest positive number (0.195), slightly surpassing the Ocra method (0.192). The two methods showed greater consistency in the selection of Slice 1, this behavior is evidenced in Figure 50a and Table 34. In addition, other methods also presented a high average, however, with a greater variation in their results, as it is the case of the Aras, Marcos, Moora, and Topsis methods. The adjustments applied to the response variable for the LM and GLM models can be seen in Figure 52.

For the Tukey's test, and by analyzing Figure 51a, the formation of two clusters was observed, showing that between the methods that make up each cluster there are no significant differences in the accuracy of slices selection, considering a confidence level of 95%. For example, for the first cluster, it is observed that i) there are no differences in performance between the Mabac, Copras, Topsis, Moora and Edas methods, which had their performance surpassed by the methods of the second cluster, and that ii) there is no significant difference in performance between them (Marcos, Codas, Aras and Ocra methods, respectively).

For Slice 2, the methods show greater variability in their results, as it can be seen in Figure 50b. This more homogeneous performance among all methods is shown in Figure 49, which consists of the Tukey test output. From the 182 possible comparisons, 58 displayed no significant difference between the methods, and this behavior is illustrated by the cluster density. Based on the estimated coefficients in the regression model, the Cocoso (3.75) and Vikor (2.66) methods demonstrated a slight superiority in relation to

Figure 49 – Behavior of the NSSF DAF methods - TS 01: Experiment 3

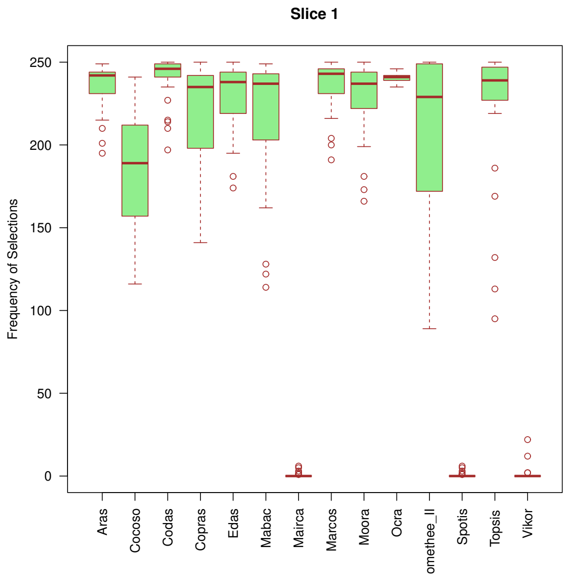


Source: elaborated by the author.

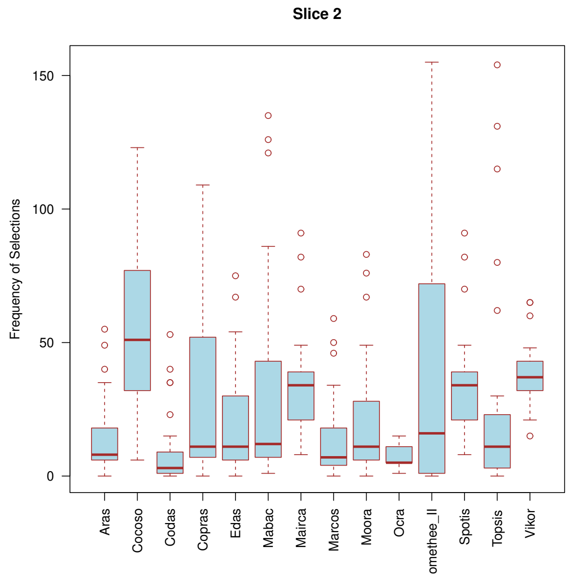
the other methods. The adjustments applied to the response variable for the LM and GLM models can be seen in Figure 53.

For Slice 3, and considering the result of the analysis of the estimated coefficients in the regression model, the Spotis (14.46), Mairca (14.46) and Vikor (14.31) methods demonstrate superiority in the selection accuracy for Slice 3. As in the other experiments, these methods had difficulties in dealing with a greater variability of the criteria used in the slice selection process, and prioritized the slice with the lowest attenuation of these parameters, that is, Slice 3. The performance of these methods is shown in Figures 50c and 49, whereas the output of the LM and GLM models are illustrated in Figure 54.

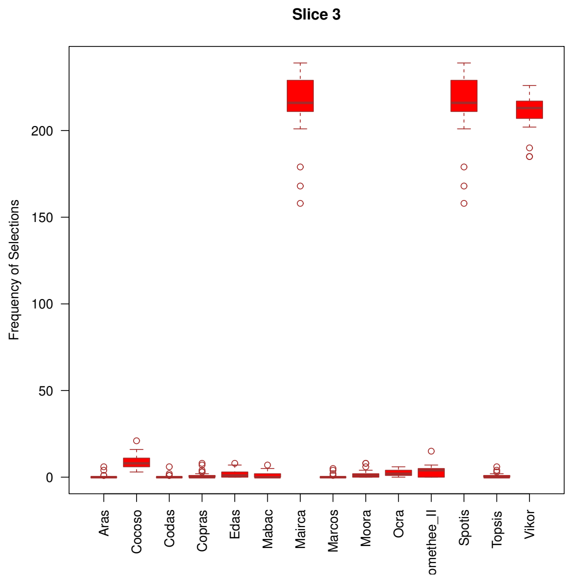
Figure 50 – Descriptive analysis using boxplot - TS 01: Experiment 3.



(a) Slice 1



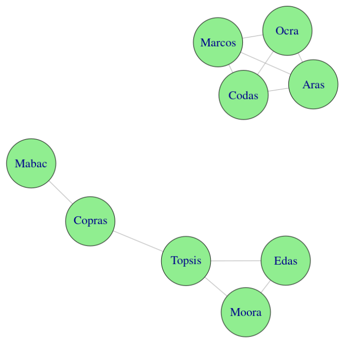
(b) Slice 2



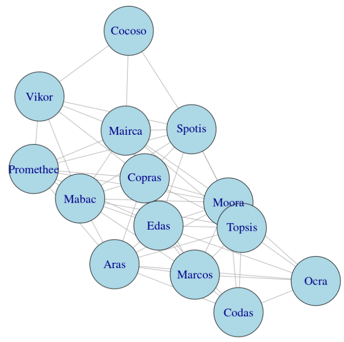
(c) Slice 3

Source: elaborated by the author.

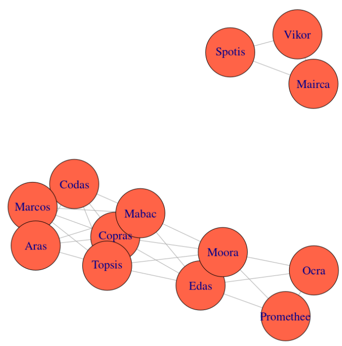
Figure 51 – Network graph TS01: Experiment 3. There is no significant difference between this methods.



(a) Slice 1



(b) Slice 2

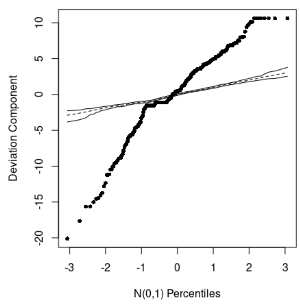


(c) Slice 3

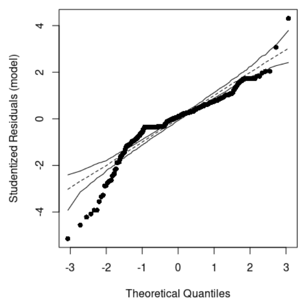
Legend: 95% confidence intervals comparing each pair of methods.

Source: elaborated by the author.

Figure 52 – Simulated envelope TS01: Experiment 3 - Slice 1.



(a) Binomial GLM.



(b) Normal LM.

Source: elaborated by the author.

Figure 53 – Simulated envelope TS01: Experiment 3 - Slice 2.

(a) Binomial GLM. (b) Normal LM.

Source: elaborated by the author.

Figure 54 – Simulated envelope TS01: Experiment 3 - Slice 3.

(a) Binomial GLM. (b) Normal LM.

Source: elaborated by the author.

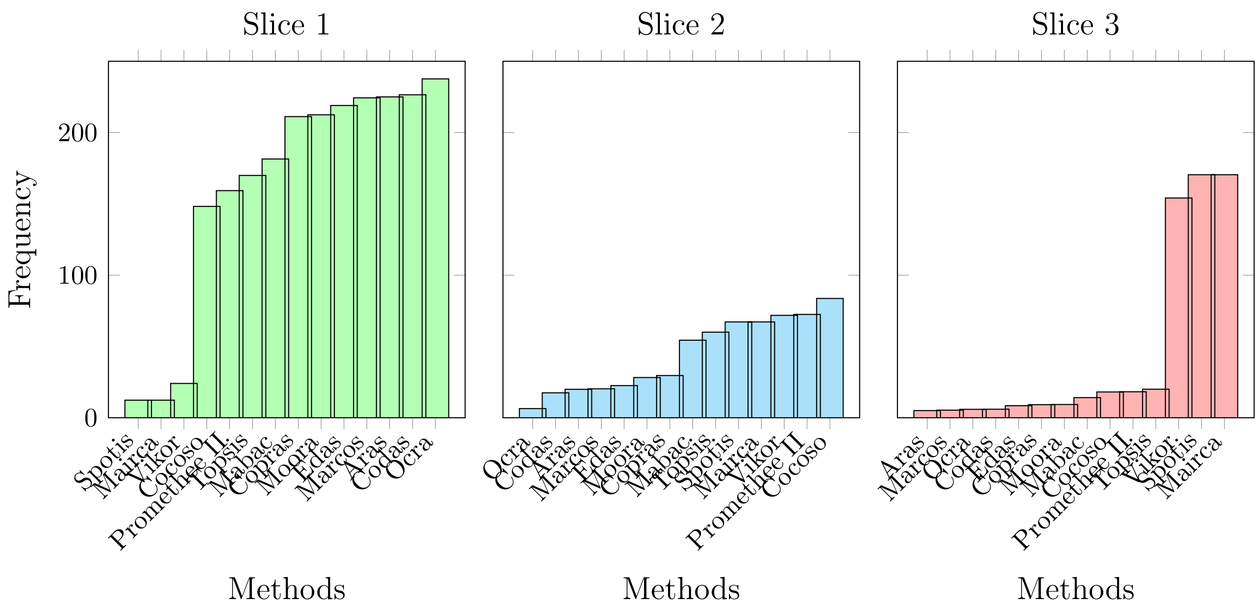
Table 34 – Methods Results: TS01 - Experiment 3

Slice	Methods	Mean	STD	Var	CI
Slice 1	Aras	234.82	14.09	198.47	229.82 – 239.81
	Cocoso	184.21	34.85	1214.67	171.85 – 196.57
	Codas	240.79	13.22	174.67	236.10 – 245.47
	Copras	220.45	31.23	975.44	209.38 – 231.53
	Edas	228.48	20.69	427.95	221.15 – 235.82
	Mabac	216.79	39.01	1521.98	202.95 – 230.62
	Mairca	0.61	1.43	2.06	0.09 – 1.11
	Marcos	235.79	15.46	239.11	230.30 – 241.27
	Moora	228.30	23.08	532.84	220.12 – 236.49
	Ocra	240.55	2.71	7.32	239.59 – 241.50
	Promethee II	200.85	53.44	2856.20	181.90 – 219.80
	Spotis	0.61	1.43	2.06	0.09 – 1.11
	Topsis	224.15	39.80	1584.32	210.04 – 238.27
	Vikor	1.15	4.30	18.51	0.37 – 2.68
Slice 2	Aras	14.82	14.12	199.40	9.81 – 19.83
	Cocoso	56.73	32.06	1028.02	45.36 – 68.09
	Codas	8.88	13.25	175.67	4.18 – 13.58
	Copras	28.45	31.53	994.38	17.27 – 39.64
	Edas	19.76	20.23	409.13	12.59 – 26.93
	Mabac	32.12	38.51	1482.98	18.47 – 45.78
	Mairca	34.12	18.73	350.80	27.48 – 40.76
	Marcos	13.85	15.48	239.63	8.36 – 19.34
	Moora	20.18	22.69	514.90	12.14 – 28.23
	Ocra	7.03	3.57	12.78	5.76 – 8.30
	Promethee II	46.06	51.26	2627.50	27.88 – 64.24
	Spotis	34.12	18.73	350.80	27.48 – 40.76
	Topsis	25.12	38.94	1516.11	11.31 – 38.93
	Vikor	38.24	10.68	114.13	34.45 – 42.03
Slice 3	Aras	0.36	1.25	1.55	0.08 – 0.81
	Cocoso	9.06	3.71	13.75	7.75 – 10.38
	Codas	0.33	1.11	1.23	0.06 – 0.73
	Copras	1.09	1.97	3.90	0.39 – 1.79
	Edas	1.76	2.18	4.75	0.98 – 2.53
	Mabac	1.09	1.65	2.71	0.51 – 1.67
	Mairca	215.27	18.35	336.70	208.77 – 221.78
	Marcos	0.36	1.14	1.30	0.04 – 0.77
	Moora	1.52	2.21	4.88	0.73 – 2.30
	Ocra	2.42	1.80	3.25	1.78 – 3.06
	Promethee II	3.09	3.42	11.71	1.88 – 4.30
	Spotis	215.27	18.35	336.70	208.77 – 221.78
	Topsis	0.73	1.35	1.83	0.25 – 1.21
	Vikor	210.61	9.24	85.31	207.33 -213.88

Legend: STD: Standard Deviation; Var: Variance ; CI: Confidence Intervals.

Source: elaborated by the author.

Figure 55 – Behavior of the NSSF DAF methods - TS 01: Experiment 4



Source: elaborated by the author.

B.3 TS-01: Experiment 4

Table 35 presents the results for Experiment 4, and Figure 55 summarizes the average performance of the methods in each slices. As occurred in the other experiments of this test setup, all parametric tests failed to accept the null hypotheses test previously defined in equations (6.2),(6.3), and (6.4).

For Experiment 4, a fair distribution of weights among all criteria was defined, according to Table 8. This configuration let the NSSF DAF choose which slice will receive network traffic from the UPF. Thus, the objective is to verify whether there are significant differences in the selection accuracy between the evaluated methods.

The same analysis was performed in this experiment. It was observed that, by using the non-parametric KRUSKAL-WALLIS test, the median of the evaluated methods are different, and the Ocra method presents the lowest IQR. Furthermore, due to the impossibility of using parametric tests, a linear regression model was implemented, and the output of the LM model was summarized in an ANOVA, enabling the application of multiple comparison methods.

For Slice 1, among the evaluated methods, the accuracy of the Ocra, Codas and Aras methods stands out from the others. When analyzing the estimated coefficients in the regression model, it appears that the Ocra method reached the highest positive number (0.424), slightly surpassing the others. Furthermore, the method had the least variability, as reported by the KRUSKAL-WALLIS test. Other methods had also an interesting

Table 35 – Methods Results: TS01 - Experiment 4

Slice	Methods	Mean	STD	Var	CI
Slice 1	Aras	225.03	15.65	244.97	219.48 – 230.58
	Cocoso	148.24	28.65	820.75	138.08 – 158.40
	Codas	226.52	19.59	383.76	219.57 – 233.46
	Copras	211.21	25.88	669.80	202.04 – 220.39
	Edas	219.00	22.96	527.25	210.86 – 227.14
	Mabac	181.48	44.02	1938.07	165.87 – 197.09
	Mairca	12.33	13.81	190.73	7.43 – 17.23
	Marcos	224.36	17.55	307.86	218-14 – 230.59
	Moora	212.48	29.81	888.76	201.91 – 223.06
	Ocra	237.67	2.30	5.29	236.85 – 238.48
	Promethee II	159.30	39.30	1544.28	145.37 – 173.24
	Spotis	12.33	13.81	190.73	7.43 – 17.23
	Topsis	169.94	56.28	3167.18	149.98 – 189.89
	Vikor	24.06	22.09	488.00	16.22 – 31.89
Slice 2	Aras	19.94	16.45	270.68	14.11 – 25.77
	Cocoso	83.67	26.42	697.79	74.30 – 93.03
	Codas	17.48	18.74	351.20	10.84 – 24.13
	Copras	29.61	23.74	563.81	21.19 – 38.02
	Edas	22.55	20.14	405.44	15.41 – 29.69
	Mabac	54.39	39.50	1559.93	40.39 – 68.40
	Mairca	67.21	38.29	1465.80	53.64 – 80.79
	Marcos	20.30	17.59	309.53	14.06 – 26.54
	Moora	28.21	25.98	674.86	19.00 – 37.42
	Ocra	6.42	2.77	7.69	5.44 – 7.41
	Promethee II	72.48	36.78	1353.13	59.44 – 85.53
	Spotis	67.21	38.29	1465.80	53.64 – 80.79
	Topsis	60.03	51.23	2624.53	41.86 – 78.20
	Vikor	71.82	48.02	2305.53	54.79 – 88.84
Slice 3	Aras	5.03	4.24	17.97	3.53 – 6.53
	Cocoso	18.09	7.35	53.96	15.49 – 20.70
	Codas	6.00	5.78	33.44	3.95 – 8.05
	Copras	9.18	7.44	55.34	6.54 – 11.82
	Edas	8.45	6.26	39.19	6.23 – 10.67
	Mabac	14.12	14.28	203.98	9.06 – 19.19
	Mairca	170.45	37.61	1414.26	157.12 – 183.79
	Marcos	5.33	3.94	15.54	3.94 – 6.73
	Moora	9.30	7.91	62.53	6.50 – 12.11
	Ocra	5.91	1.97	3.90	5.21 – 6.61
	Promethee II	18.21	11.48	131.80	14.14 – 22.28
	Spotis	170.45	37.61	1414.26	157.12 – 183.79
	Topsis	20.03	22.18	492.03	12.16 – 27.90
	Vikor	154.12	50.80	2580.61	136.11 – 172.13

Legend: STD: Standard Deviation; Var: Variance ; CI: Confidence Intervals.

Source: elaborated by the author.

performance, as in Marcos, Moora and Edas methods. However, they presented a greater variance in their results. These inferences can be seen in Table 35, and in Figures 56a and 55.

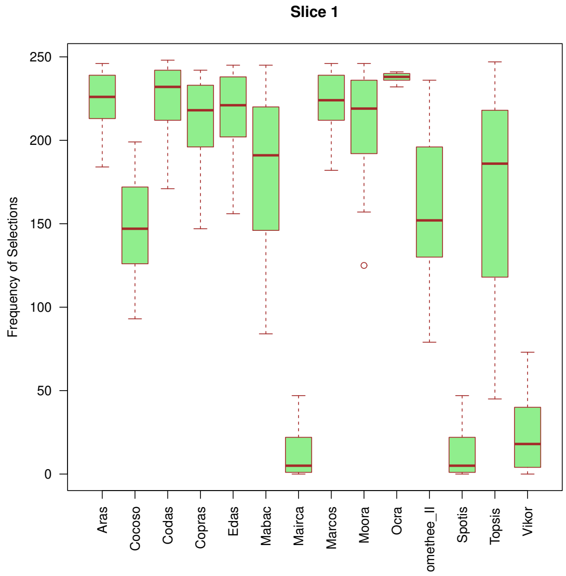
The comparisons output performed in the Tukey's test to Slice 1 shows that in 30 out of the 182 comparisons there are no significant difference between the methods. These comparisons can be visualized in Figure 57a. Note that 03 clusters are formed, two of which are linked through the Mabac method, since the performance of this method is close to the group represented by the third quartile and the maximum value obtained by the methods, as shown in Figure 56a. The Mairca and Spotis methods showed the worst performance, a characteristic that was repeated in the other tests. The distribution analysis for the LM and GLM models are shown in Figure 58.

For Slice 2, and considering the estimated coefficients analysis of the regression model, the Cocoso (5.08) and Promethee II (4.25) methods showed a slight superiority over the other methods. This performance is depicted in Figure 56b and Figure 55. In the comparison performed from Tukey's method, we observed that in 39 comparisons there were no significant differences between the methods, considering a confidence level of 95%. These comparisons can be visualized in Figure 57b, which can also be observed in the presence of two clusters, which represent the dispersion conferred by the boxplot graph of Figure 56b.

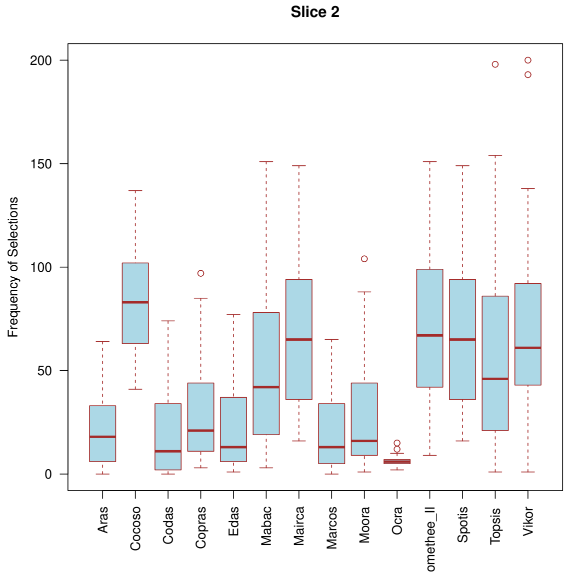
For Slice 3, using the descriptive analysis data shown in Figure 56c, and considering the estimated coefficients in the regression model, the Spotis (10.94), Mairca (10.94) and Vikor (10.17) methods demonstrate superiority in selection accuracy. This performance is totally different from the other MCDM methods, and demonstrates the difficulty of these methods in scenarios where the variability of the criteria values are high, making the updates of their decision matrices ineffective, and, therefore, changing their ranking power. At the output of the Tukey's test, it was found that in 38 comparisons there was no significant difference between the methods, as illustrated in Figure 57c. Note that the Vikor, Mairca and Spotis methods form a cluster, while most of the methods have a very similar performance, with a slight superiority for the Cocoso, Promethee II, Topsis and Mabac methods that address the first part of the second cluster.

Adjustments in the distribution of response variable for the slices in Experiment 4 can be seen in Figures 58, 59, and 60.

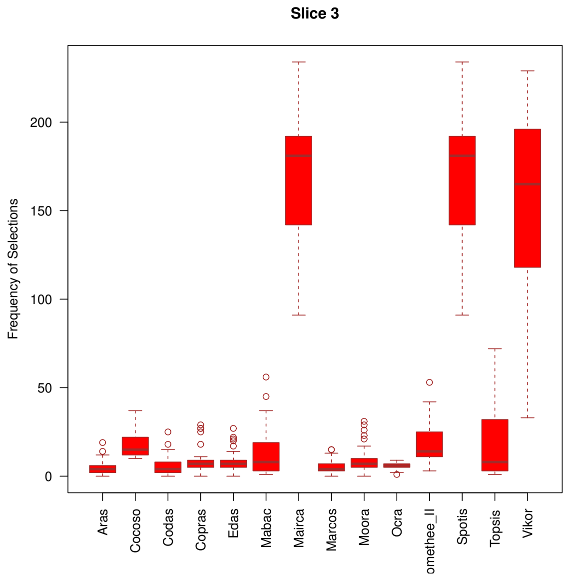
Figure 56 – Descriptive analysis using boxplot - TS 01: Experiment 4.



(a) Slice 1



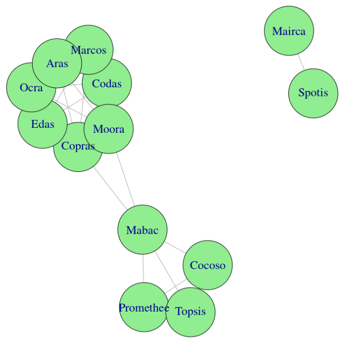
(b) Slice 2



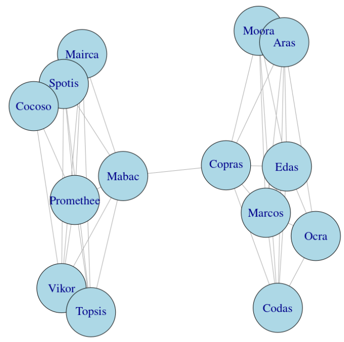
(c) Slice 3

Source: elaborated by the author.

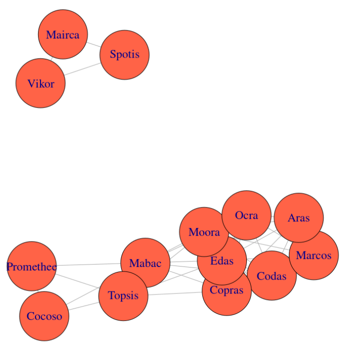
Figure 57 – Network graph TS01: Experiment 4. There is no significant difference between this methods.



(a) Slice 1



(b) Slice 2

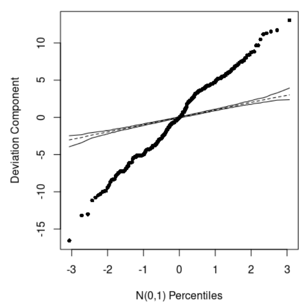


(c) Slice 3

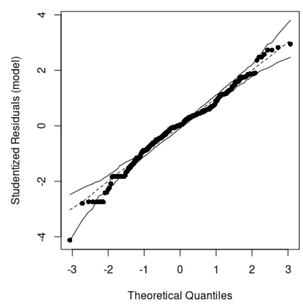
Legend: 95% confidence intervals comparing each pair of methods.

Source: elaborated by the author.

Figure 58 – Simulated envelope TS01: Experiment 4 - Slice 1.



(a) Binomial GLM.



(b) Normal LM.

Source: elaborated by the author.

Figure 59 – Simulated envelope TS01: Experiment 4 - Slice 2.

(a) Binomial GLM. (b) Normal LM.

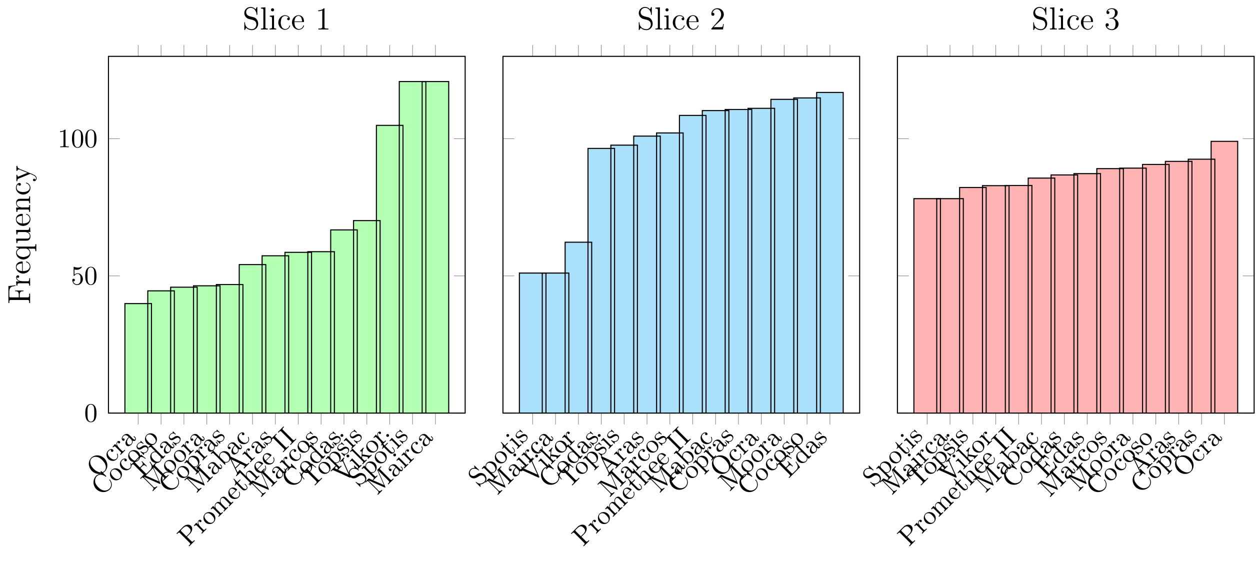
Source: elaborated by the author.

Figure 60 – Simulated envelope TS01: Experiment 4 - Slice 3.

(a) Binomial GLM. (b) Normal LM.

Source: elaborated by the author.

Figure 61 – Behavior of the NSSF DAF methods - TS 03: Experiment 2



Source: elaborated by the author.

B.4 TS-03: Experiment 2

Results for Experiment 2 are reported in Table 36 and summarized in Figure 61. As in Experiment 1, in the analysis of the response variable (Frequency of Selections) for all slices, there was rejection of the null hypothesis, i.e., the data does not come from a normal distribution, does not show independence of residues and homoscedasticity of variances. From the KRUSKAL-WALLIS test, the rejection of the null hypothesis was obtained for the main question conducted in this study and defined in the equation (6.1). In fact, the methods show difference in performance for the evaluated scenario.

For Slice 1, the use of the vector of weights defined in Table 7, improved a performance gain in almost all methods compared to Experiment 1, according to Figure 62a and Table 36. The accuracy of the methods occurs due to the improvement in the differentiation of attributes and in the judgment of preferences, which increases the operational capacity of the methods, mainly in the construction of the ranking of alternatives. The exception occurs again for the methods Mairca, Spotis and Vikor, which despite the use of weights, they had slightly lower performance than in Experiment 1 for the same slice. Even so, from analyzing the estimated coefficients of the regression model, these methods exhibited the best results, namely: Spotis (3.51), Mairca (3.51) and Vikor (2.75).

After applying the Tukey's method to the results obtained in Slice 1, it was found that, for 36 comparisons, there was no significant difference between the methods. These

Table 36 – Methods Results: TS03 - Experiment 2

Slice	Methods	Mean	STD	Var	CI
Slice 1	Aras	57.33	18.09	327.17	50.92 – 63.75
	Cocoso	44.55	5.71	32.57	42.52 – 46.57
	Codas	66.76	19.16	367.06	59.96 – 73.55
	Copras	46.85	17.28	298.76	40.72 – 52.98
	Edas	45.88	17.04	290.48	39.84 – 51.92
	Mabac	54.12	12.41	153.92	49.72 – 58.52
	Mairca	120.82	14.11	199.03	115.82 – 125.82
	Marcos	58.82	17.89	320.22	52.47 – 65.16
	Moora	46.39	16.26	264.50	40.63 – 52.16
	Ocra	39.91	10.20	104.02	36.29 – 43.53
	Promethee II	58.58	13.02	169.56	53.96 – 63.19
	Spotis	120.82	14.11	199.03	115.82 – 125.82
	Topsis	70.15	15.27	233.07	64.74 – 75.56
	Vikor	104.85	13.93	194.13	99.91 – 109.79
Slice 2	Aras	100.94	27.38	749.93	91.23 – 110.65
	Cocoso	114.85	10.39	108.01	111.16 – 118.53
	Codas	96.45	33.37	1113.32	84.62 – 108.29
	Copras	110.64	29.13	848.61	100.31 – 120.97
	Edas	116.85	29.15	849.63	106.51 – 127.18
	Mabac	110.24	18.93	358.25	103.53 – 116.95
	Mairca	51.03	14.18	200.97	46.00 – 56.06
	Marcos	102.09	27.01	729.27	92.52 – 111.67
	Moora	114.33	28.44	808.79	104.25 – 124.42
	Ocra	111.06	18.87	356.12	104.37 – 117.75
	Promethee II	108.48	18.47	341.26	101.93 – 115.03
	Spotis	51.03	14.18	200.97	46.00 – 56.06
	Topsis	97.64	22.16	490.99	89.78 – 105.49
	Vikor	62.27	13.25	175.58	57.57 – 66.97
Slice 3	Aras	91.73	17.67	312.08	85.46 – 97.99
	Cocoso	90.61	6.88	47.37	88.17 – 93.05
	Codas	86.79	19.21	368.92	79.98 – 93.60
	Copras	92.52	18.44	329.88	85.97 – 99.05
	Edas	87.27	17.91	320.77	80.92 – 93.62
	Mabac	85.64	10.75	115.49	81.83 – 89.45
	Mairca	78.15	11.26	126.76	74.16 – 82.14
	Marcos	89.09	15.64	244.52	83.55 – 94.64
	Moora	89.27	17.67	312.33	83.00 – 95.54
	Ocra	99.03	14.24	202.78	93.98 – 104.08
	Promethee II	82.94	9.56	91.31	79.55 – 86.33
	Spotis	78.15	11.26	126.76	74.16 – 82.14
	Topsis	82.21	11.62	134.92	78.09 – 86.33
	Vikor	82.88	7.16	51.30	80.34 – 85.42

Legend: STD: Standard Deviation; Var: Variance ; CI: Confidence Intervals.

Source: elaborated by the author.

comparisons are illustrated in the network graph in Figure 63a.

For Slice 2, there is greater variability of the response variable, this is indicated by the number of outliers, as illustrated in Figure 62b. This behavior occurs due to the features inherent to the slice itself, which holds the specifications with hight computational resources and enable a high throughput, thus serving a larger number of UEs, as detailed in Tables 5 and 6. In general, the performance of the methods follows a more homogeneous pattern, and from the multiple comparisons provided from Tukey's method, it was observed that in 56 pairwise comparisons there was no significant difference between the methods. However, based on the estimated coefficients analysis of the regression model, the Edas (0.76), Cocoso (0.74) and Moora (0.65) methods are slightly superior.

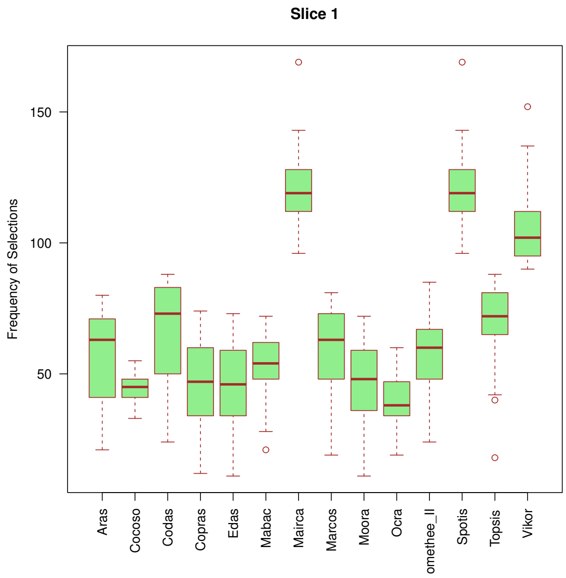
Slice 3 follows the same pattern as the other slices, and shows greater homogeneity in the results of the methods, as seen in Figure 62c. From analyzing the estimated coefficients of the regression model, it was observed that basically all methods had similar performance, with the Ocra method (0.39) showing a slight performance gain. The multiple comparisons reported from Tukey's method show that in 78 comparisons there was no significant difference between the methods. This result is demonstrated in Figure 63c, in which there is only one cluster that presents a higher density in its connections.

In general, the results of this experiment are similar to those of Experiment 1. However, a slight performance gain is perceived due to the application of the weight vector, i.e., the differentiation of attributes improves the ranking capacity of the methods, and, consequently, the accuracy of slices selection process. The analysis of response variable (Frequency of Selections) considering a binomial distribution, and the correction applied by $sqrt()$ function into the LM model to approximate a normal distribution, for all slices, can be observed in Figures 64, 65 and 66.

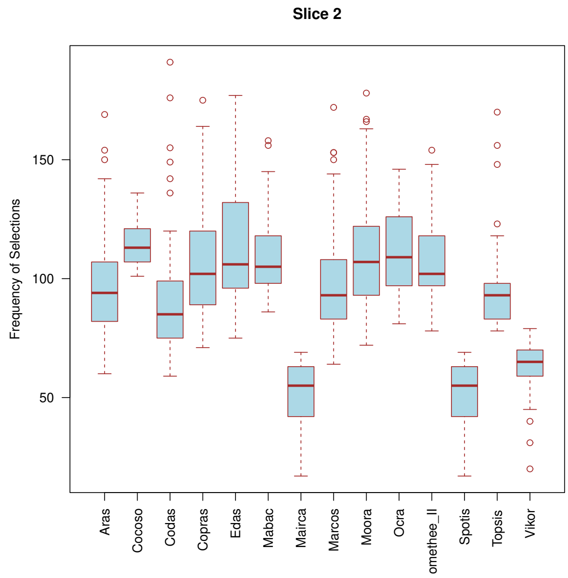
B.5 TS-03: Priority data

This experiment adopted the weights defined in Table 7. For this, the following criteria order was considered: *"Loss", "Distance", "Download", "Reliability", "Density", "Latency", "Upload","Jitter"*. The same strategy used to analyze the results in Section 6.4 was adopted here. In this sense, the tests defined in the hypotheses of the equations (6.2), (6.3) and (6.4) were carried out. In addition, we sought to validate the hypothesis of the equation (6.1), which aims to verify whether there were significant differences in the performance of the methods for high priority traffic.

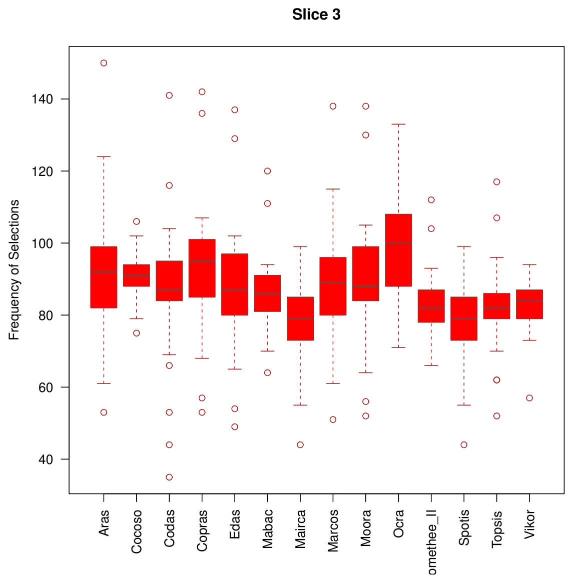
Figure 62 – Descriptive analysis using boxplot - TS 03: Experiment 2.



(a) Slice 1



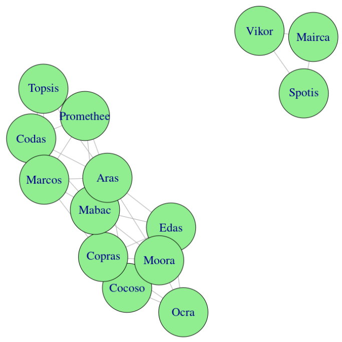
(b) Slice 2



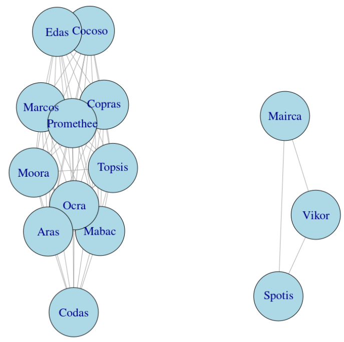
(c) Slice 3

Source: elaborated by the author.

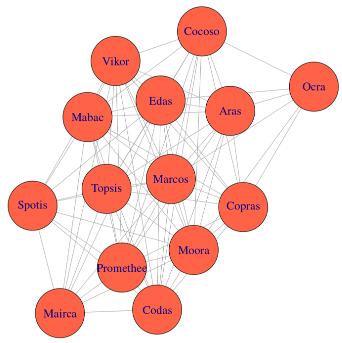
Figure 63 – Network graph TS03: Experiment 2. There is no significant difference between this methods.



(a) Slice 1



(b) Slice 2

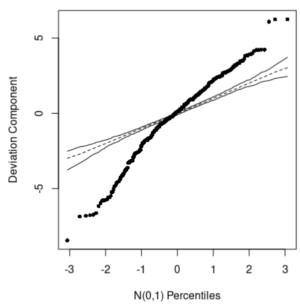


(c) Slice 3

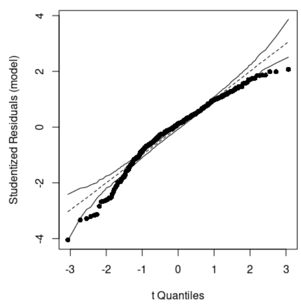
Legend: 95% confidence intervals comparing each pair of methods.

Source: elaborated by the author.

Figure 64 – Simulated envelope TS03: Experiment 2 - Slice 1.



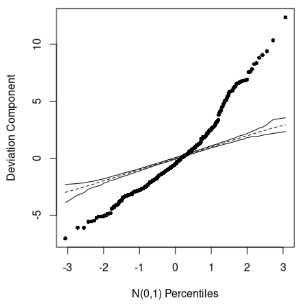
(a) Binomial GLM.



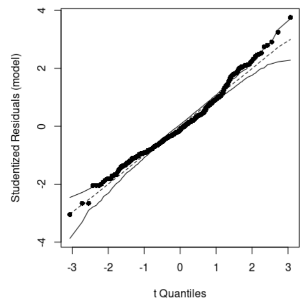
(b) Normal LM.

Source: elaborated by the author.

Figure 65 – Simulated envelope TS03: Experiment 2 - Slice 2.



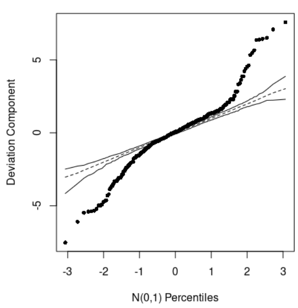
(a) Binomial GLM.



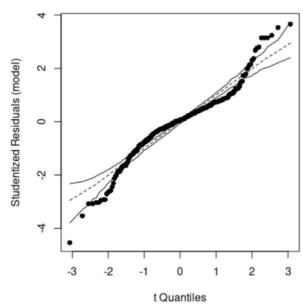
(b) Normal LM.

Source: elaborated by the author.

Figure 66 – Simulated envelope TS03: Experiment 2 - Slice 3.



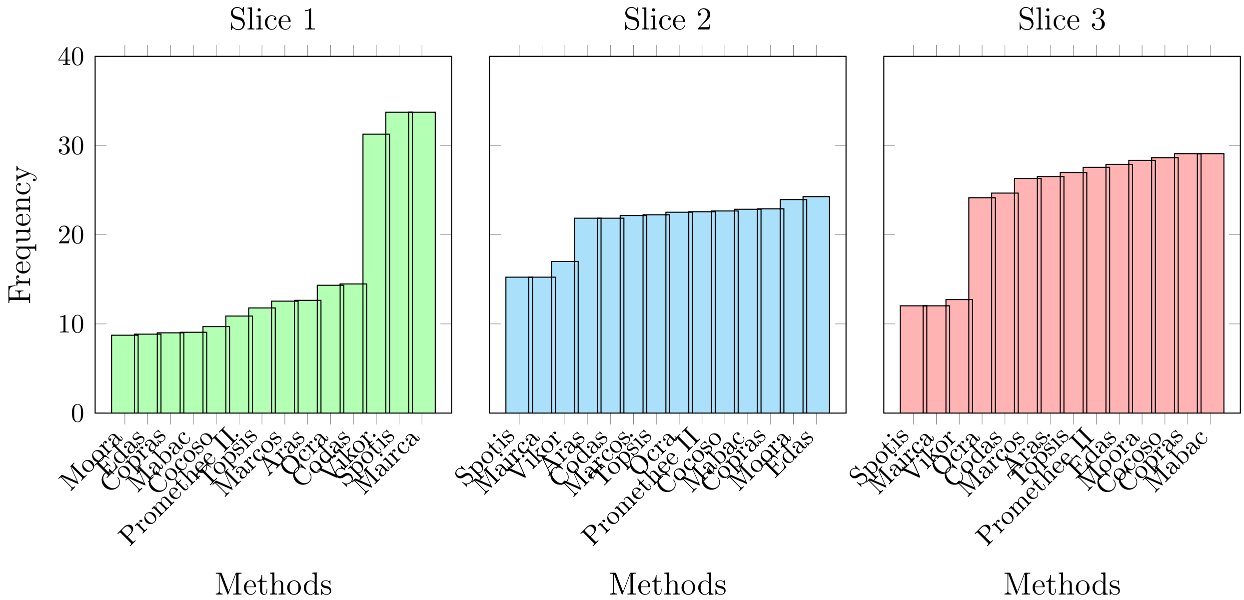
(a) Binomial GLM.



(b) Normal LM.

Source: elaborated by the author.

Figure 67 – Behavior of the NSSF DAF methods - TS:03 Alarm



Source: elaborated by the author.

In this regard, to validate the proposed hybrid algorithms behavior, as well as the efficiency of the NSSF DAF architecture for critical IoT scenarios, the occurrence of 200 priority flows was considered. This event simulates the input of alarm messages for the real-time health system scenarios and Geohazard prevention.

These packages were generated using the Scapy[1] library, representing payloads compatible with these applications. Using the previously trained predictive model, and considering the MCDM methods and technologies adopted in the scenario in Figure 15, the results obtained are detailed in Table 37 and summarized in Figure 67. It is important to note that for all slices there was a violation of the premises of normality, independence of residuals and homoscedasticity of variances.

[1]https://scapy.net/

Table 37 – TS:03 Alarm - Method Results with priority data

Slice	Methods	Mean	STD	Var	CI
Slice 1	Aras	12.64	3.22	10.36	11.49 – 13.78
	Cocoso	9.70	1.59	2.53	9.13 – 10.26
	Codas	14.48	3.99	15.95	13.07 – 15.90
	Copras	9.00	3.67	13.50	7.70 – 10.30
	Edas	8.85	3.55	12.57	7.59 – 10.11
	Mabac	9.06	3.12	9.75	7.95 – 10.17
	Mairca	33.73	4.70	22.08	32.06 – 35.39
	Marcos	12.55	3.26	10.63	11.39 – 13.70
	Moora	8.73	3.54	12.52	7.47 – 9.98
	Ocra	14.33	3.32	11.04	13.16 – 15.51
	Promethee II	10.88	3.53	12.48	9.63 – 12.13
	Spotis	33.73	4.70	22.08	32.06 – 35.39
	Topsis	11.79	3.65	13.30	10.49 – 13.08
	Vikor	31.27	5.23	27.39	29.42 – 33.13
Slice 2	Aras	21.85	5.53	30.63	19.89 – 23.81
	Cocoso	22.67	2.71	7.35	21.71 – 23.63
	Codas	21.85	6.26	39.13	19.63 – 24.07
	Copras	22.91	6.11	37.27	20.74 – 25.07
	Edas	24.27	6.26	39.20	22.05 – 26.49
	Mabac	22.85	4.22	17.82	21.35 – 24.35
	Mairca	15.24	3.57	12.75	13.98 – 16.51
	Marcos	22.15	4.80	23.01	20.45 – 23.85
	Moora	23.94	5.94	35.31	21.83 – 26.05
	Ocra	22.52	6.41	41.07	20.24 – 24.79
	Promethee II	22.58	4.45	19.81	20.99 – 24.15
	Spotis	15.24	3.57	12.75	13.98 – 16.51
	Topsis	22.24	5.62	31.63	20.25 – 24.24
	Vikor	17.00	4.30	18.50	15.47 – 18.52
Slice 3	Aras	26.52	5.97	35.70	24.40 – 28.63
	Cocoso	28.64	2.88	8.30	27.61 – 29.66
	Codas	24.67	6.38	40.67	22.41 – 26.93
	Copras	29.09	6.11	37.34	26.92 – 31.26
	Edas	27.88	5.68	32.23	25.87 – 29.89
	Mabac	29.09	5.26	27.71	27.22 – 30.96
	Mairca	12.03	2.52	6.34	11.14 – 12.92
	Marcos	26.30	5.28	27.84	24.43 – 28.17
	Moora	28.33	5.73	32.79	26.30 – 30.36
	Ocra	24.15	5.27	27.82	22.28 – 26.02
	Promethee II	27.55	4.54	20.63	25.93 – 29.16
	Spotis	12.03	2.52	6.34	11.14 – 12.92
	Topsis	26.97	5.70	32.53	24.95 – 28.99
	Vikor	12.73	2.38	5.64	11.89 – 13.57

Legend: STD: Standard Deviation; Var: Variance ; CI: Confidence Intervals.

Source: elaborated by the author.

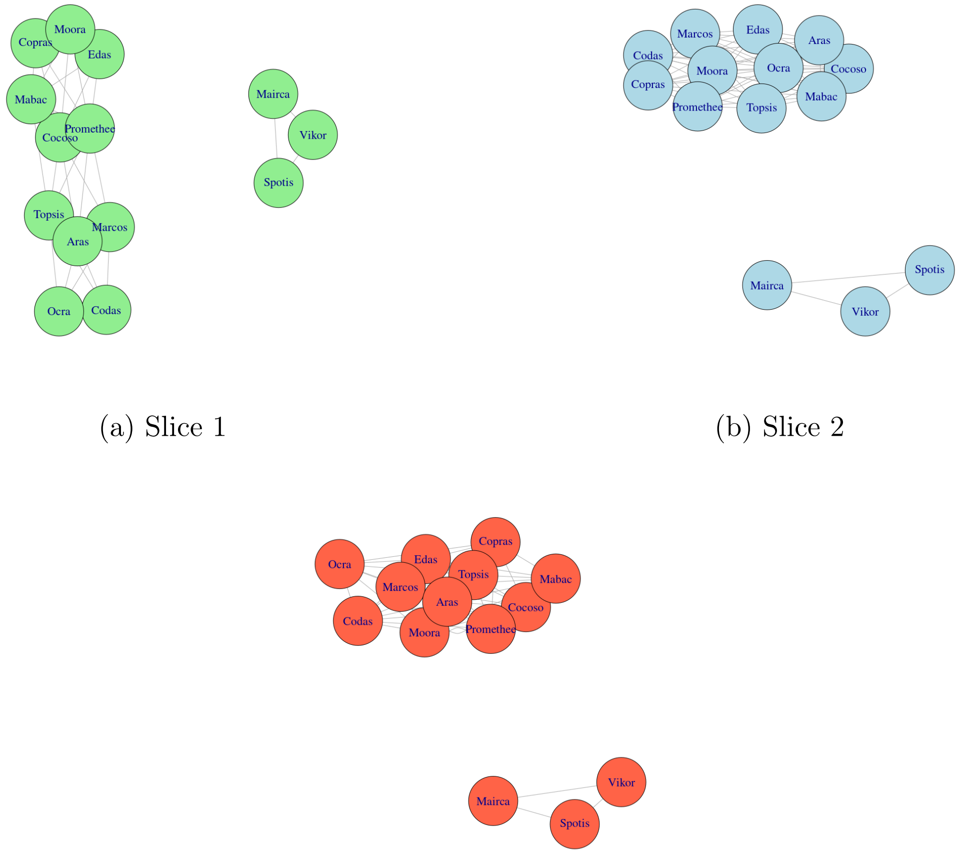
For Slice 1, it was observed that the Aras, Cocoso and Vikor methods had the lowest IQR, demonstrating a more homogeneous distribution of the response variable (Frequency of Selections). In addition, by using the KRUSKAL-WALLIS test, it was found that there are differences in the accuracy of method selection. From the estimated coefficients analysis of the regression model, there is evidence of the superiority of the Mairca (2.26), Spotis (2.26) and Vikor (2.05) methods. The Tukey's test showed that in 35 comparisons there is no significant difference between the methods. This behavior is represented in Figures 69a. In the network graph of Figure 68a, the formation of 03 clusters can be observed, two of which are connected, whereas the other, formed by the Mairca, Spotis and Vikor methods, is isolated. This characteristic, as previously mentioned, is recurrent, and shows the divergence of these methods in comparison to the others.

In Slice 2, there is a more homogeneous performance of the methods; however, the estimated coefficients analysis of the regression model shows a slight superiority of the Edas (0.24) and Moora (0.21) methods, as shown in Figure 69b and 67. From the Tukey's Test it was verified that in 58 pairwise comparisons there is no significant difference between the methods. The network graph in Figure 68b reports the equivalence of the methods, with an exception for the Spotis, Mairca and Vikor methods, which, in this slice, presented a lower performance than the other methods.

For Slice 3, and considering the results obtained by the estimated coefficients of the regression model, there is evidence of a slight superiority of the Mabac (0.25) and Copras (0.24) methods, as shown in Table 37 and Figure 69. In general, the methods had a similar performance. According to the comparisons carried out by the Tukey's test, and considering a confidence level of 95%, it was verified that in 53 comparisons there is no significant difference in performance between the methods, as illustrated in Figure 68c, through the represented density in the cluster.

The experiment showed that even in situations involving priority traffic, the MCDM methods maintain the consistency of judgments of the alternatives, and are sensitive to the attenuation of the traffic conditions of the slices. This characteristic is evidenced by the lower variability of the response variable, when compared with the results discussed in Section B.4 for the three slices. In addition, there was a balance in the delivery of priority flows between the evaluated slices, with Slice 1 receiving 61 flows (30.5%) of priority traffic), Slice 2 receiving 72 flows (36%) and Slice 3 receiving 67 flows (33.5%). It is important to note that there was no change in the treatment of the queues from UPF function, i.e., the judgment and direction occurred only through the processing carried out in the NSSF DAF.

Figure 68 – Network graph TS03: Alarm. There is no significant difference between this methods.



(a) Slice 1

(b) Slice 2

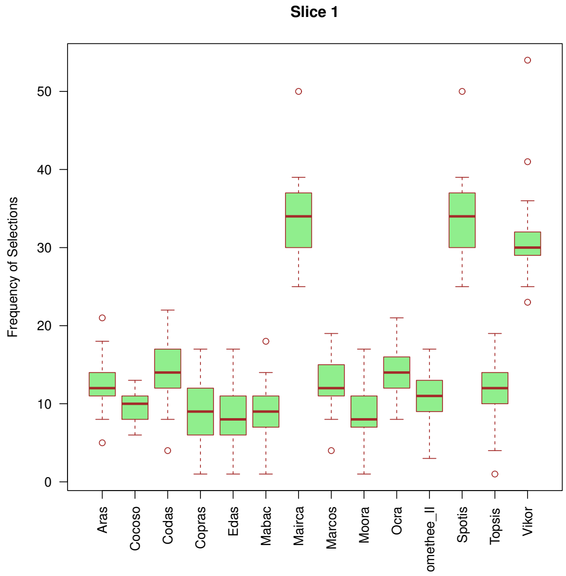
(c) Slice 3

Legend: 95% confidence intervals comparing each pair of methods.

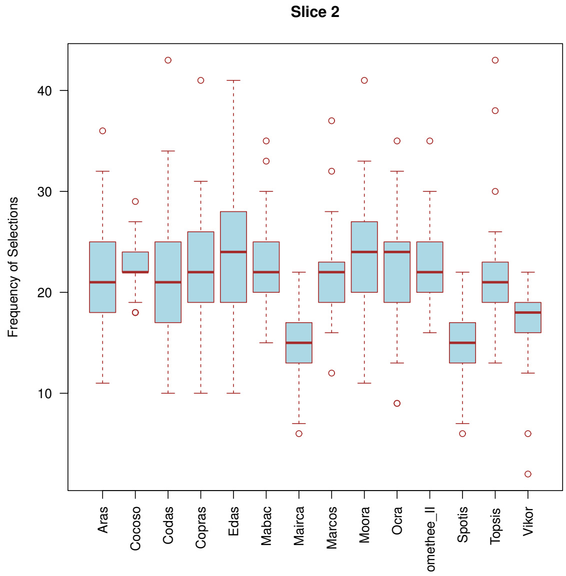
Source: elaborated by the author.

The simulated envelope graphs for this experiment can be seen in Figures 70, 71 and 72. We highlight the adjustment performed by the LM model with the $sqrt()$ function on the response variable for Slice 2.

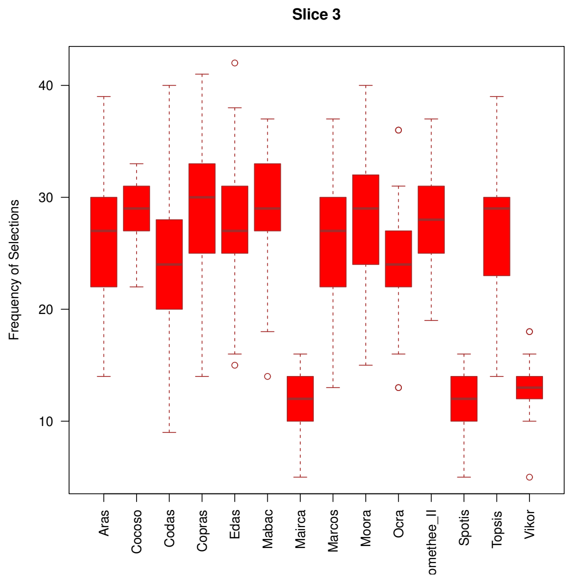
Figure 69 – Descriptive analysis using boxplot - TS: 03 Alarm



(a) Slice 1



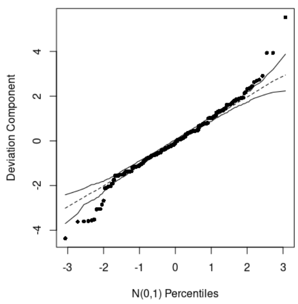
(b) Slice 2



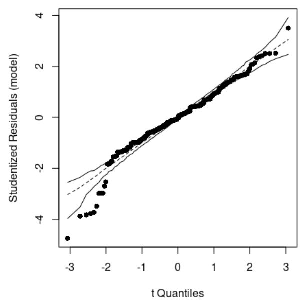
(c) Slice 3

Source: elaborated by the author.

Figure 70 – Simulated envelope TS03: Alarm - Slice 1.



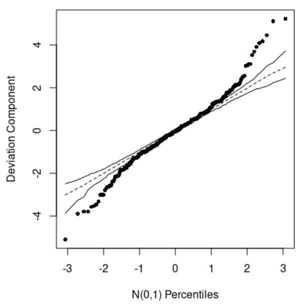
(a) Binomial GLM.



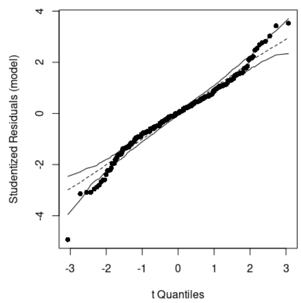
(b) Normal LM.

Source: elaborated by the author.

Figure 71 – Simulated envelope TS03: Alarm - Slice 2.



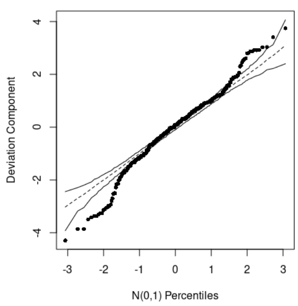
(a) Binomial GLM.



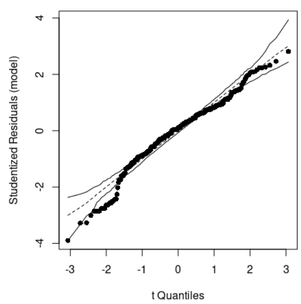
(b) Normal LM.

Source: elaborated by the author.

Figure 72 – Simulated envelope TS03: Alarm - Slice 3.



(a) Binomial GLM.



(b) Normal LM.

Source: elaborated by the author.