Noisy Neighbour Impact Assessment and Prevention in Virtualized Mobile Networks

Francisco Muro, Eduardo Baena, Sergio Fortes, *Member, IEEE*, Lars Nielsen, and Raquel Barco

***Abstract*—The generalization in the use of virtualization in the upcoming generation of cellular networks involves new paradigms and approaches for their management. The correct sharing of the underlying resources between multiple virtualized network functions as well as any other processes sharing the same computational platform implies a complex architecture that makes virtualization challenging. As a result, numerous variables (e.g., computational capacity) were mostly ignored by previous management systems, but that can now lead to relevant impacts on the service performance. In this virtualized scenario with multiple coexistent processes over the same hardware, a "Noisy Neighbour" (NN) is identified as an entity that uses most of the underlying resources while other virtual units suffer a lack of them. While difficult to identify, such situations can affect the network service. In this context, the present work analyzes and assesses the NN problem for 5G Core scenarios. A complete emulated 5G network and analysis framework is defined and developed to evaluate the impact of a noisy entity. In this way, the degradation that Key Performance Indicators suffer in the network and by the end-users when a NN appears is assessed. Thus, the present work proposes a baseline for handling NN through a novel lifecycle management flow. For this purpose, it is evaluated the effectiveness of multiple Machine Learning (ML) models for the identification of NN, based on the metrics gathered from the proposed framework, achieving 99% of accuracy. Moreover, ML is applied to develop a method for network performance inference, along with a prediction model to forecast the number of CPU resources the network may demand at any given time supporting the proposed management flow.**

***Index Terms*—Software defined networking, 5G, network function virtualization, network slicing, Noisy Neighbour, machine learning.**

Manuscript received 13 April 2022; revised 15 July 2022; accepted 17 July 2022. Date of publication 27 July 2022; date of current version 7 March 2023. This work has been partially funded by: Ministerio de Asuntos Económicos y Transformación Digital y la Unión Europea - NextGenerationEU within the framework "Recuperación, Transformación y Resiliencia y el Mecanismo de Recuperación y Resiliencia" under the project MAORI; Junta of Andalucía and the ERDF in the context of the project IA^2MON-5G - "Inteligencia Artificial para el Análisis y Monitorización de Redes de Comunicación 5G" (Ref. UMA-CEIATECH-13, "Proyecto singular de actuaciones de transferencia del conocimiento CEI Andalucía TECH. Ecosistema innovador con inteligencia artificial para Andalucía 2025") and postdoctoral scholarship (Ref., DOC_01154, "selección de personal investigador doctor convocado mediante Resolución de 21 de mayo de 2020", PAIDI 2020). The associate editor coordinating the review of this article and approving it for publication was M. F. Zhani. *(Corresponding author: Eduardo Baena.)*

Francisco Muro, Eduardo Baena, Sergio Fortes, and Raquel Barco are with the Telecommunication Research Institute, E.T.S. Ingeniería de Telecomunicación, Universidad de Málaga, 29010 Málaga, Spain (e-mail: fmc@ic.uma.es; ebm@ic.uma.es; sfr@ic.uma.es; rbm@ic.uma.es).

Lars Nielsen is with Keysight Technologies, Santa Rosa, CA 95403 USA (e-mail: lars.m.nielsen@keysight.com).

Digital Object Identifier 10.1109/TNSM.2022.3194137

I. INTRODUCTION

IN RECENT years, the development of cellular networks has shown the evolutionary trend toward more agile, resilient, and efficient deployments. In this regard, the growing volume of data traffic, the flexibility requirements for new services, and the search for greater efficiency in capital expenditure (CAPEX) and operational expenditure (OPEX) have made virtualization an essential property for future telecommunication networks. This saves costs by running on general-purpose computing servers the elements that until recently were hardware-specific, such as routers, switches, etc. In addition, it is easy to create new virtual machines (VM) when a service is saturated or to quickly configure and deploy a new service that the operator wants to offer on its infrastructure. Standard organizations like ETSI or 3GPP are defining the mechanisms for virtualizing mobile network infrastructures. As a result, there is a proposal for the virtualization of network functions (Network Function Virtualization, NFV), network services (composed of several VNFs), and automating their lifecycle (instantiate them, monitor their performance, scale them, etc.) through the Network Function Virtualization Management and Orchestration framework (Management and Orchestration or MANO). From this perspective, one of the most interesting applications that enable the use of NFV in mobile networks is Network Slicing [1], [2], [3], [4], where a virtualized network infrastructure can be divided into "slices" [5], [6], [7], [8], [9], [10], which are offered to different types of service. Each of these segments must manage the virtual resources so that they do not interfere with each other, offering the service the perception of exclusive use of the underlying infrastructure. Thus, the main difficulty in distributing computing resources to enable NS relies on satisfying two objectives: on the one hand, it is necessary to ensure independence between slices (functional isolation), allowing their adaptability to meet service requirements. On the other hand, it is required to achieve flexibility and adaptability of the common resources shared between the different slices. Therefore, the application of NS will be crucial to meet the divergent requirements demanded by upcoming services. Due to the complex development and evaluation of this functionality, previous works about NS are performed through simulators and testbeds formed of each network element. As such, several projects like [11] and [12] are representative of this kind of study.

Even when the NS capability becomes available, one of the critical aspects of managing slices is the concept of Noisy

Neighbour (NN). This term identifies one problem that arises when computing resources are shared among several VMs, containers, or services in a pool, and one consumes enough to deny the other entities the level of service they expect to receive. This situation causes performance failures in other virtual units that require those resources and, consequently, causes a degradation of the service of virtualized mobile networks.

Firstly, the identification of the presence of an NN is not a trivial process, depending on multiple factors that determine the state of the network and its underlying architecture, and is therefore performed through strategies that require the use of artificial intelligence algorithms [13]. Likewise, the heterogeneity of the different situations that may evidence the appearance of a NN also involves a complex process of extracting and analyzing network metrics. To such an extent that the existing literature does not provide up-to-date analyses evaluating 5G Core (5GC) performance in terms of throughput, latency, and packet loss, related to the existence of a NN.

Secondly, although it has been the subject of study in several works, there is no uniqueness in the literature about the process of handling a NN. The preferred solution proposed is the migration of the impacted virtual unit (VM or container), which implies a cost in terms of downtime and performance degradation [14]. Given the fact that the new generation of mobile networks intends to support the sharing of the same physical infrastructure between operators and satisfy the requirements demanded by different applications, a more complex operational context than their predecessors is envisioned. Thus, the NN problem represents one of the leading research topics in mobile network virtualization, given its devastating impact, which could cause significant losses to operators and service providers.

So far, there is no test environment in the literature where the impact of the Noisy Neighbour in 5G, explicitly focusing on the core, has been demonstrated. There are also no previous studies on which parameters most influence the consumption of virtualized resources at different load levels. Although there are references to VNF management [15], [16], [17], [18], there are no specific works dealing with NN when it affects 5G core functions. And finally, while there are some suggestions for algorithms that identify NN or predict VNFs performance, they have not been described as part of a 5G production environment. Therefore, the present work presents the following contributions:

1) Development of an innovative Noisy Neighbour (NN) emulation testbed in a 5GC scenario including the necessary tools for modeling and evaluation.
2) Application of factorial design (within the Design of Experiments framework) for the evaluation of the impact of 5GC service features versus resource consumption.
3) Design of a novel lifecycle management flow for NN (NN-LCM) that establishes a baseline in the 5GC context. All relevant features as well as possible solutions in an NFV environment have been considered.
4) Implementation and evaluation of multiple ML algorithms that improve the accuracy of the NN identification task beyond SoA, achieving more than 99% of accuracy.
5) Application of several ML forecast models that enable the proposed NN-LCM.

Including the present introduction section, the paper is organized as follows: Section II presents a description of the State of the Art concerning the NN issue and the applications of AI-based methods for handling this entity. Then, Section III describes the proposed framework to model the NN environment, the tools, and the methodology adopted. In addition, Section IV exposes the assessment of the test campaign, analyzing the impact of the input variables. This section also presents the evaluation of the performance degradation caused by the NN. Following, Section V describes the framework for NN management in a 5G environment as well as the results of the mechanisms required for its implementation: NN identification and prediction mechanisms for decision making. Finally, Section VI exposes the key findings achieved.

II. State of the Art

The virtualization of the network resources and their dynamic assignment, commonly performed by a hypervisor, is a critical component for the new generation networks. Thus, it has been a subject of study of numerous works that intend to evaluate the scalability of the deployed services over a virtualized network. The objective of this management is to avoid both the under-use of the resources, which will cause inefficient use of the infrastructure and the over-provisioning, which will increase CAPEX and OPEX for network operators.

In this context, the study given in [19], in which NVF-based LTE-EPC (Long Term Evolution-Evolved Packet Core) framework to implement end-to-end (E2E) "slices" is proposed in order to give support to the dynamic resource assignment for the involved Data Planes (DPs), managed by an autoscaling developed mechanism. Others, such as the work in [20] focus their study on the performance and scalability analysis in the Control Plane of NS-enabled 5G networks, based on the session establishment and user registering processes.

Another research line in virtualization is assessing the number of resources demanded by a virtual unit in charge of running a VNF. Regarding this matter, in [21], the behaviour of the packet processing of a VNF is modeled to study the number of resources that a VNF needs to run the task properly. Over this, a prediction model is also proposed to avoid over-demand situations by running a resource scaling and/or the migration of the virtual unit.

The NN has been evaluated in different ways in virtualized environments. The most common method consists of the employment of a Cloud Computing infrastructure to deploy a NN scenario. Over such infrastructure, several VMs are deployed on this infrastructure: one running the evaluated VNF and another one running a "noisy" process, which is a high-consumption process that demands a large number of CPU resources ([22]). Since the assignment of resources is dynamic, the VMs that contain the "noisy" process will make use of most of the available resources and cause a NN behaviour.

In contrast, the present work uses a novel framework that intends to deploy the NN in the same VM in charge of running the VNF in order to abstract the underlying infrastructure study as well as having a complete control of the available resources for the evaluated VNFs, enabling the possibility of extracting a wide range of network behaviours in the presence of a differently scaled NN.

When a NN appears, depending on its impact on the service and the context, involves the management of the virtualized environment to mitigate its effects. This is a key aspect that has been addressed from a general point of view by the VNF lifecycle management. This involves the development and evaluation of algorithms that intend to predict and address the network's resources needs dynamically by deploying VNF instances over the virtualized infrastructure. The works given in [17], [23], [24] and [25] provide a model of these solutions.

Despite the importance of optimizing the deployment and the management of the active VNF instances for addressing network load demands, no contribution approaches the VNF lifecycle management while considering the resources competition problems caused by any anomalous VNFs' behaviour, such as the one addressed in this work, the NN, what is regarded as a critical challenge in virtualized deployments.

Numerous studies have been conducted on the identification of this entity through the application of techniques based on artificial intelligence, supported by Machine Learning (ML) models, such as Support Vector Machine (SVM) and Random Forest (RF), addressed in [13] and Deep Learning, such as Convolutional Neural Networks (CNNs), applied in [26]. Other works focus on the evaluation of their possible causes (Root Cause Analysis, RCA) [22]. On the other hand, the results in [7], [27] address the assessment and management of the NN problem. In the latter, a model for identification and reduction of the impact of the virtual unit identified as NN is proposed through a mechanism based on CPU pinning and load balancing built on the NS support, which is posed as an alternative to the traditional migration of the affected virtual machine (or container), due to its high computational cost and time spent.

III. Design and Implementation of the 5GC NN Scenario

In this section, the development of the 5G Core Noisy Neighbour scenario is explained as well as the NN simulator implementation as a crucial component. Then, Design of Experiments (DoE) is introduced as the methodology to obtain the inputs combinations to optimize the number of test executions in order to deliver reliable and statistically significant results. DoE also supports the later analysis derived from the performed test campaign.

A. Environment Modelling

The 5G network environment with NS support can be modeled in multiple ways through virtual machines, containers, or processes that execute the network functions of the 5GC. In this work, the tool that supports the 5G environment is LoadCore 5G Core Testing [28], which allows recreating a complete network through the independent simulation of each of the network nodes: Radio Access Network (RAN), 5GC (Standalone) and Data Network (DN).

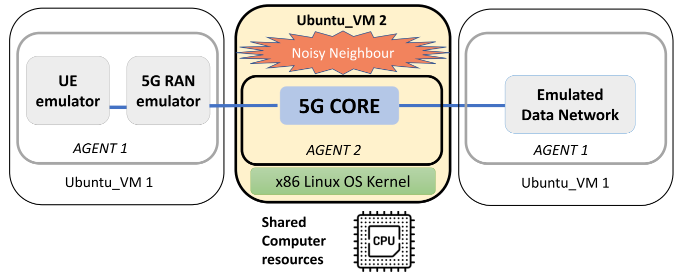


Fig. 1. 5GC Noisy Neighbour evaluation scenario.

TABLE I
NOISY NEIGHBOUR MODEL NOTATION

P_i	Physical Machines (PM) (i ∈ {1, ..., N })
V_{ij}	Virtual Machines/Containers (j ∈ {1, ..., n_i })
C_{ik}	Total capacity of resource k (k ∈ {1, ..., K })
U_{ik}	Utilization factor of resource k in P_i (k ∈ {1, ..., K })
Θ	Threshold of usage
T	Time period considered significant
$D_{ijk}(t)$	Capacity request of k type by V_{ij} at time t

The emulation of the network components is done through the execution of an agent. An agent is a process running on a Linux system, capable of developing the functionality of the different nodes. For the system implementation, two agents are available, running on different VMs:

- *Ubuntu_VM 1:* the machine in charge of running the Agent 1 process, which emulates the 5G network endpoints (User Equipments (UEs), RAN and DN);
- *Ubuntu_VM 2:* the machine in charge of running the Agent 2 process, which emulates all the VNFs that compose the 5GC.

The objective is to include a NN attached to the VM in charge of running the Agent 2 process, reducing the number of resources available for the 5GC's VNFs and, thus, causing performance degradation in the 5G core Figure 1.

Although there are multiple definitions of the NN concept in the literature depending on the field of application [27], [22], [29], [30], in this paper the NN is identified as an entity that blocks access to a shared pool of computing resources P_i for a significant period of time *(T)* to the rest of the VNFs that are hosted in it.

In this respect, as shown in Table I, a resource pool consisting of N physical machines is considered. A number n_i of virtual machines denoted as V_{ij} are hosted on the P_i machine. Also, the capacity of type k of the P_i machine is denoted C_{ik} (for example $k = 1$ stands for vCPU, $k = 2$ refers to Memory and $k = 3$ denotes Storage).

In this environment, each container or virtual machine generates a resource demand D_{ijk} at each time instant that depends in turn on service characteristics $S(t)$ (i.e., packet rate, packet size) as well as workload (depending on the number of users) $W(t)$ as described below.

$$D_{ijk} \equiv D_{ijk}(W(t), S(t)) \tag{1}$$

The condition that the resource demand at a given instant τ cannot exceed the available capacity is satisfied according to the following expression:

$$\sum_{j=0}^{n_i} D_{ijk}(\tau) \leq C_{ik} \tag{2}$$

The utilization factor U_{ik} is thus defined as the ratio of demand to capacity. When this ratio is low, the demand for resources is met by the available capacity (non-competitive environment) as follows:

$$U_{ik} = \frac{\sum_{j=0}^{n_i} D_{ijk}}{C_{ik}} \tag{3}$$

Thus, the VNF denoted as V_{ix} is considered a "Noisy Neighbour" if, during an interval time of T duration $t \in (t_z, t_{z+1}, \ldots t_{z+T})$ it exceeds a utilization threshold fulfilling the following conditions:

$$U_{ik} > \Theta \tag{4}$$

$$D_{ixk} >> \sum_{j \neq x}^{n_i} D_{ijk} \tag{5}$$

Since resource demand varies as a function of time-dependent processes $W(t)$ and $S(t)$, it is crucial to be able to predict the value of the effect these will have in order to determine whether this is a NN or, on the contrary, a normal peak in VNF operation.

B. Noisy Neighbour Simulator

In the related literature, NN modeling and monitoring are performed in different ways, depending on the infrastructure supporting the scenario. A common choice is often to create "noise" in the system by introducing an expensive computational process that executes a stressful task and demands a large amount of CPU resources. Also, such processes are deployed on VMs that share resources with the VM in charge of executing the VNF under evaluation [27].

However, this work proposes a novel NN model by deploying the NN on the same VM running the 5GC to abstract the environment from the underlying virtualization infrastructure. Thus, the 5GC VNFs and the NN will coexist on the same VM, with the latter being responsible for limiting the CPU resources available to the 5GC while deployed on a Linux system. Such an environment would be similar to containerized deployments such as Docker.

One aspect worth highlighting is the fact the time of use of CPU resources in Linux systems relays on the scheduler. In this way, from version 2.6.2 of Linux kernel implements the Completely Fair Scheduler (CFS) scheduler in order to ensure fairness in resource sharing. Thus, as the processes that are used in the present work are run on top of an Ubuntu (Linux) system the method of generating a NN with a high resources demand thread is not considered, since the presence of the scheduler avoids a full control of the limits imposed by such implementation of the NN.

Therefore, based on the Linux "cpulimit" daemon, a Python script capable of modeling a NN has been developed. This tool allows setting the maximum value of the CPU resources percentage that a process can take by controlling its performance through sending signals (SIGSTOP and SIGCONT) to that process. In this way, it is possible to directly determine the CPU use time of the 5GC, fixing the maximum amount of resources available at each moment. This enables the control of the NN scale and simulates the effect caused by a virtual unit run on top of a virtualization platform (or Cloud Computing), controlling most of the available resources. Thus, the NN simulator takes four arguments as inputs to set its functionality described in Table II.

TABLE II
INPUT ARGUMENTS FOR THE NN SIMULATOR

Argument number	Name	Description
1, 2	cpu_low_limit, cpu_high_limit	Sets the values range among the ones that the NN will choose the CPU resources limit for the agent
3	cpulimit_restart_time	If this value is set to -1, the NN enables standard mode. If it is higher than 0, it will enable persistence mode.
4	time_between_lectures_for_logging	Time between monitoring lectures that will be recorded in a file

Below, the operation of the NN simulator is presented sequentially:

1) Finds the process running the LoadCore Agent and captures its PID.
2) Launches a thread that executes the "cpulimit" tool to limit the agent's CPU usage to a fixed value. This value will be chosen randomly in a range indicated by the user.
3) Launches a sub-process to perform the monitoring, recording in a file the CPU usage information for the processes involved in the environment: the 5GC (Agent 2) and the NN.
4) The script provides two modes of use to increase its versatility:
   - *Standard mode:* the thread in charge of executing the NN (through "cpulimit") maintains its execution until user's interruption.
   - *Persistence mode:* a restart period (T) is established so that, every T seconds, the thread that executes the NN will be restarted with a new CPU usage limit, randomly chosen in the range specified by the user when launching the NN.

The developed NN simulator allows identifying the process that runs the 5GC's VNFs, randomly choosing a maximum CPU usage value and limiting its resources to the chosen value. In this way, the environment of an NN is modeled, in which it consumes most of the resources of the underlying architecture reducing the performance of the rest of the processes.

C. Test Campaigns

1) Design of Experiments: The design of experiments models are statistic classical models based on the experimentation that aim to determine if some input factors influence a feature of interest. Traditionally, the experimentation process is restricted by the time and/or the number of tests that can be run. In this context DoE helps to determine an optimal number of measurements depending on the ranges of each parameter to detect the most significant changes in the system response. Thus, in the 5GC NN scenario, the use of this methodology

TABLE III
INPUT PARAMETERS FOR THE TEST CAMPAIGN

Input parameters	Description	Number of levels	Levels
Number of users	Number of users interacting through upstream traffic injection	2	50, 65535
Payload size (bytes)	Data payload for packets injected into the network	3	40,300,1280
Packet rate (pkt/s)	Number of packets per second injected into the network	3	60, 5000, 35000

TABLE IV
MEASURED KPIS

Network output	Description	Measuring tool
Outbound throughput (Mbps)	User plane traffic in the 5GC	LoadCore.
End-to-end delay (ms)	Time elapsed by the packets from the time they are sent to the time they are received. A time series will be extracted in which, periodically, the information is provided in the form of a histogram	LoadCore
Packet Loss (pkts/s)	Number of packets lost in the network as a result of excessive load on the 5GC	LoadCore VNFs
CPU usage (%)	CPU resources used by the 5GC at each test time	NN Simulator

allows to synthetically find out the different network states achieved depending on the combinations of available input parameters.

A factorial design has been used to select the combinations of levels for each of the input parameters. Since the objective of the present work is to evaluate the impact of the network over the User Plane traffic and according to the employed tool capabilities, the traffic flow used as input for a simulation will be constant upload UDP traffic (injected from the UEs to DN). Both the considered parameters and the proposed levels according to the factorial design are summarized in Table III.

Through the combinations of input variable levels obtained by factorial design [31], 18 experiments (Factorial Design 2x3x3) are defined, to be run in two independent campaigns: (1) Campaign without NN; (2) Campaign with NN.

As network output, for each performed test, the Key Performance Indicators (KPI) of the network are extracted, as well as those coming from the VNFs. The outputs that constitute the datasets obtained are listed in Table IV.

D. Data Pre-Processing

Each test launched through the emulated 5G network consists on three phases:

1) *Phase 1:* User registering process.
2) *Phase 2:* Data transmission.
3) *Phase 3:* User de-registering process.

During phases 1 and 3, there is no data sent by the users, and the amount of resources that the 5GC requires and the engagement of each VNF through the implied processes are highly different from Phase 2. Thus, according to the objective of the present work, Phases 1 and 3 have not been considered for the present study.

On the other hand, each of the experiments has been labeled with a Boolean value (0 or 1) to determine the existence or not of a NN. These labels are assigned depending on the settings established for a specific emulation to enable the application of supervised learning models.

As a final result, an unified dataset with 8822 rows and columns has been obtained containing the inputs and the outputs of the network previously exposed.

IV. INFLUENCE ANALYSIS AND THE NOISY NEIGHBOUR IMPACT

In this section, first the factorial design analysis is presented to find which of the input parameters has a higher correlation with the 5GC CPU resource consumption. Accordingly, two main objectives are defined to achieve this goal:

1) Identifying which type of services has a higher impact to the resource consumption of a 5GC. The heterogeneous use cases defined for 5G arise the need of characterizing the different services in terms of their resource consumption, what is achieved through this study.
2) Determining which of the inputs has a higher interaction with a response of interest during the launching of a test campaign. This is considered as one of the most relevant contributions due to the limited amount of time to which a campaign is subjected, allowing to reduce the number of needed tests.

Secondly, the impact of the NN on 5GC performance is demonstrated by comparing the collected KPIs in an emulation test affected by the NN with another one running normally. The aim is to evaluate the degradation induced by a NN on the network KPIs. Such phenomenon would manifest itself in a reduction of the Quality of Service (QoS) provided to users, causing serious losses to network operators and service providers.

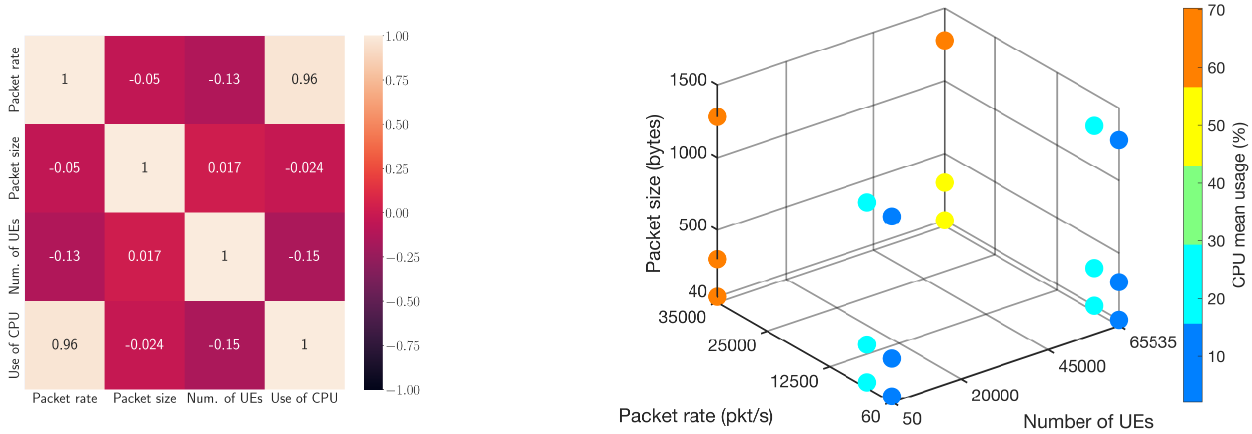
A. Influence Analysis

As discussed above, the influence analysis aims to characterize the resource consumption of the core as a function of the network inputs. A correlation heatmap between the proposed variables is initially depicted in Figure 2a.

The correlation between upstream packet rate and 5GC resource demand can be observed to be close to 1 (0.96), revealing a high relationship between both characteristics and proportional behaviour between packet rate and computing resources consumption. In contrast, the rest of the input variables are significantly less correlated with CPU usage.

Likewise, to verify the premise obtained from the previous analysis, Figure 2b presents the three dimensions corresponding to the emulation inputs (number of users, packet size and packet rate), while the fourth dimension is presented by using a specific colour for mean CPU consumption.

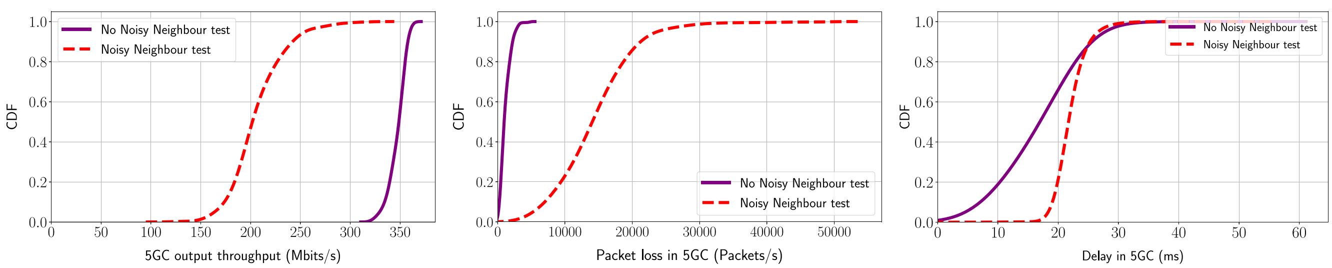
In this way, Figure 2b evidences the proportionality between packet rate and the 5GC CPU usage, which overcomes the 70% for the emulation launched with 35000 packets/s of packet rate.



(a) Correlation heatmap of inputs and CPU resources

(b) Average CPU utilization of 5GC

Fig. 2. Analysis of the influence of input parameters.



(a) Throughput

(b) Packet Loss

(c) E2E Delay

Fig. 3. Experimental CDFs per KPI scenario comparison.

It is possible to infer that not those services that demand high throughput, but those that involve sending high packet rates (with a high independence of packet size) will lead to a greater demand for 5GC computational resources.

Thus, services such as massive machine type communications (mMTC) or real-time interactive services, which involve sending short packets with a high packet rate, will require a large provision of computing resources.

B. Noisy Neighbour Impact

In the previous section, the analysis of a 5GC in a not impacted by a NN scenario has been presented. Hence, the hypothesis is that a NN can cause a degradation of the impact against the network performance indicators due to the lack of available resources for the 5GC.

To verify this assumption, two different scenarios have been envisaged: (a) a scenario with no resource constraint for 5GC and (b) another one suffering the resource constraint due to the presence of a NN.

The selected test with 35000 packets/s packet rate, 65535 users and 1280 bytes of packet payload represents the biggest challenge in terms of resource demand due to a high level of traffic load.

Thus, the comparison between scenarios allows evaluating the impact of NN on 5GC performance, where the obtained metrics can evidence the degradation caused by the lack of resources. The cumulative density function (CDF) of the performance indicators has been estimated using the kernel density estimation (KDE) method.

The corresponding performance and packet loss graphs for the impacted and unimpacted emulation tests are presented in Figure 3a and Figure 3b The curves illustrate how NN causes a decrease in network throughput and increases packet losses in the network due to lack of resources. In addition, the delay increases significantly, as depicted in Figure 3c.

Briefly, this section has shown the noticeable decrease in network performance in the presence of a Noisy Neighbour, caused as a consequence of the lack of resources.

This performance degradation will directly affect the QoS perceived by the user, most likely causing large losses for the operator.

V. Noisy Neighbour Management

The existence of multiple "slices" deployed over the same physical infrastructure and the independence requirements for each of them may imply a more extensive complexity in network management.

The root cause identification may not be a trivial process; it may be due to the malfunctioning of one of the virtual units or excess resource demand to satisfy the service requirements.

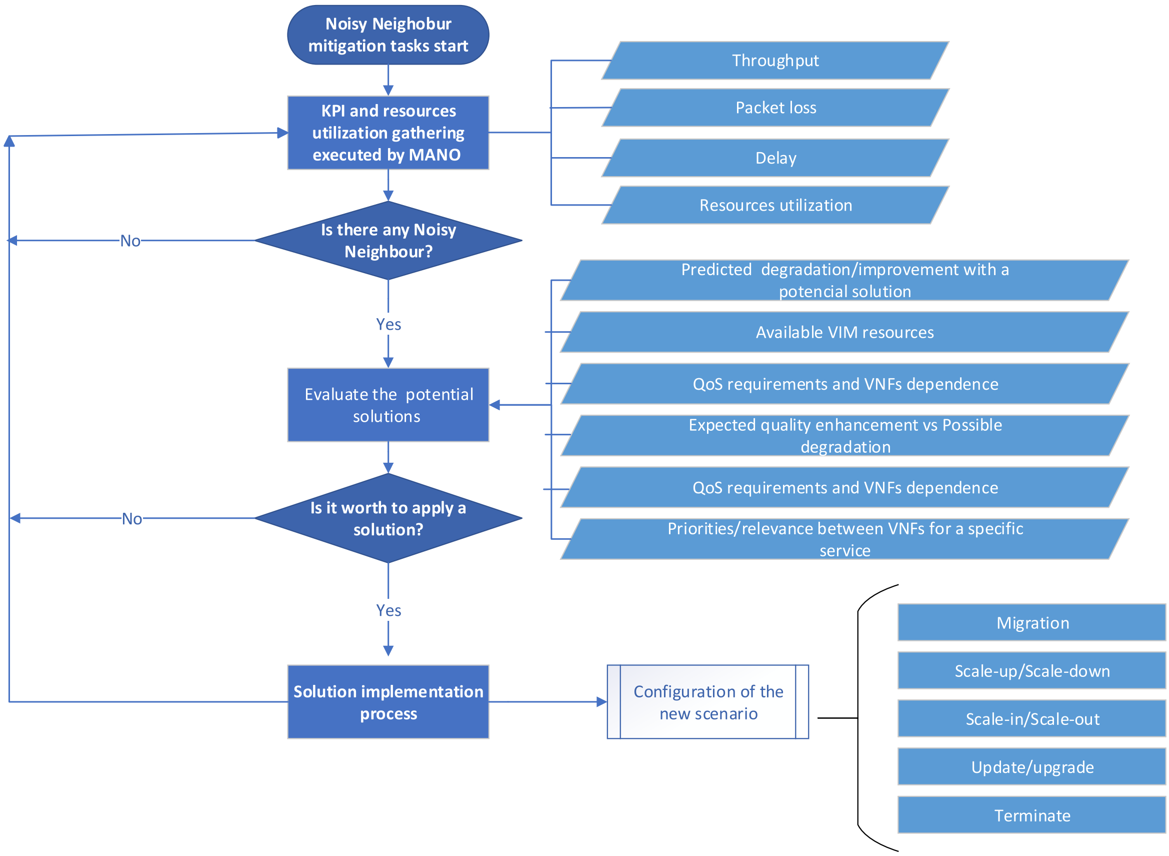


Fig. 4. Noisy Neighbour Life Cycle Management flow.

Therefore, the NN problem management (VM migration, resources fixing, or NN suppression) implies a decision-making process based on external factors involving operators and service providers, such as priorities among the "slices" or service level agreements.

However, there may appear in the system some NN that, due to the need to satisfy specific requirements, and with a priority operation over the rest of the virtual units that share the same resources, cannot be suppressed (either by fixing resources or other techniques to block NNs).

In addition, any solution involving the migration of an impacted virtual unit involves a cost, so that a decision taken to mitigate the NN problem could also have an impact on network performance and the quality of the service provided.

For all these reasons, in the particular case of the 5GC scenario, the management of the NN must be carried out with caution. In this sense, a life cycle management flow is presented (Section V-A) that allows structuring the actions from the identification (Section V-B) as well as incorporating through prediction the necessary information for an adequate decision making in each situation (Section V-C).

A. *NN Lifecycle*

Several works in the literature have assessed the importance of optimizing the MANO lifecycle management operations ([32]), which involves the scaling, migration, upgrade and termination of VNFs operating in a network [24].

Thus, as a contribution of the present work, a flowchart is presented that aims to establish a base model for Noisy Neighbour Lifecycle Management (NN-LCM) in 5GC scenario, considering the MANO monitoring tasks, the QoS requirements of the deployed services, the correlation between the performance of the VNFs and the solution impact on the quality of service. This NN-LCM flow is presented in Figure 4.

The NN-LCM flow presents MANO as the responsible entity of monitoring and performing all the tasks for evaluating the context and the potential solutions. Once a Noisy Neighbour has been identified, the task becomes more complex. As assessed above, the implementation of a solution always involves a cost: the migration of the VNF entails downtime, which could be unacceptable for an Ultra Reliable Low Latency service; conversely, the limitation of one of the entities sharing the resources to make them available for a specific VNF might not obtain the expected results as a consequence of the performance correlation.

Therefore, two models are proposed: one to predict the network performance according to the network context and available resources, and the other to forecast the expected amount of resources to obtain a full quality service.

Thus, as a preliminary step, an identification of the presence of NN would be carried out. If positive, it would be possible to evaluate the different potential solutions in terms of cost-effectiveness, and to develop cost models for evaluation and decision making based on the results of the estimation.

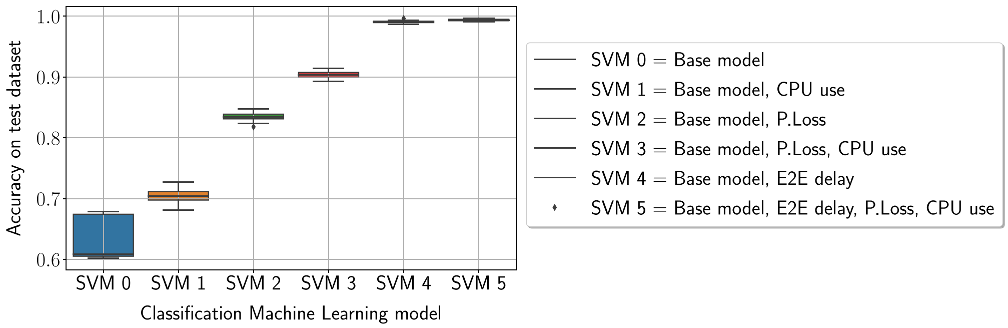


Fig. 5. KPI supply relevance comparison.

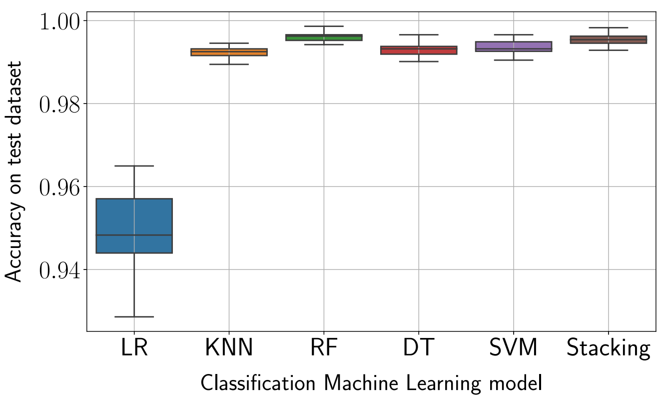


Fig. 6. Comparison of ML models for NN identification.

B. Noisy Neighbour Identification

The analysis of the results obtained shows the impact that the NN has on the network, which decreases its performance and leads to a decrease in throughput, an increase in end-to-end delay, and a higher number of lost packets.

The identification of the NN corresponds to the first step for the later decision making and the problem management in charge of mitigating its impact.

The identification of the NN is one of the critical aspects of the problem's management process. Therefore, several studies have evaluated the efficiency of different AI algorithms over distinct network scenarios to detect the entity. In this context, the identification of an NN using the Support Vector Machine (SVM) and Random Forest (RF) models was carried out in [13], where more than 90% accuracy was achieved, using the CPU load and input and output throughput as input variables for the algorithms. These results were later improved by the same authors in [26] by using a CNN, taking advantage of the temporal nature of the metrics used in the process, increasing the accuracy to more than 96%.

Taking these works as a reference, several ML classification models [33], [34] have been evaluated through the proposed scenario: Logistic Regression based classifier (LR), K-Nearest Neighbours (KNN) [35], RF, Decision Tree (DT), SVM with cubic polynomial kernel, and the stacking method, which considers all these algorithms as level-0, and LR for level-1. All the models are evaluated through a cross-validation with 3 folds and 10 repetitions over the entire dataset.

The set of tests used for the analysis contains the metrics of each emulation realization labeled with a boolean value that shows the presence of a NN in the experiment.

The results are presented in the form of a boxplot (Figure 6) showing the model accuracy. The SVM baseline enhances the accuracy to a value over 99%, which is close to a non-failure classification. This improvement in NN identification regarding the previous contributions is attributed to the increase in the number of KPIs provided to the model (with the addition of packet loss and delay metrics).

Moreover, and in order to test the importance of the features provided to the ML models for NN identification, the SVM accuracy obtained by varying the number of KPIs provided has been compared following a similar scheme to the one applied in [36]. From the "base model" where only the input and output throughput as the inputs are taken, the increase in accuracy obtained is shown as more information is added. Figure 5 reveals the noticeable enhancement in the classification score obtained by SVM through the use of campaign data, which evidences the decrease in the estimation variance and accuracy increase while providing more network information to the model.

C. Prediction Models

As previously exposed, the complexity of the virtualized 5G network involves the need of considering multiple correlated variables as the relations between VNFs or resource-aware behaviours. This increases the need to use the ML to ensure that all this information is handled and tracked. In addition, it is also necessary to provide the ML algorithms with enough data to correctly estimate the network behaviour, as reflected by the KPIs.

For this reason, two forecasting mechanisms are proposed and evaluated, in order to provide the sufficient information for the decision-making process during the NN-LCM: (a) KPI prediction (b) Required CPU resources.

1) Prediction of KPIs: The objective of the following model is to forecast the performance of the network, based on users traffic demand conditions (given by Packet rate, Packet size, Number of UE) and the CPU consumption stats of the 5GC. In this way, and in a similar fashion to previous works dedicated to the estimation of cellular network quality of experience/key quality indicators (KQI) under different radio conditions [4], [35], [37], the proposed model will enable the

TABLE V
FORECAST MODELS PERFORMANCE COMPARISON

Predicted feature	Throughput				Lost packets				Delay E2E				Use of CPU			
ML algorithm	*SVM*	*PR*	*RF*	KNN	*SVM*	*PR*	*RF*	KNN	*SVM*	*PR*	*RF*	KNN	*SVM*	*PR*	*RF*	KNN
R2	0.715	0.977	0.983	0.983	0.49	0.746	0.776	0.785	0.907	0.943	0.967	0.968	0.970	0.968	0.981	0.981
MAE	25637.68	3158.05	2910.32	3049.39	1611.50	1193.83	997.93	997.53	6739.78	1761.65	939.93	927.62	2.36	2.36	1.68	1.70
MASE (%)	80.00	4.43	99.74	4.28	80.37	25.99	98.75	21.72	90.74	12.69	100.21	6.68	95.34	8.88	96.63	6.38

possibility of forecasting the performance obtained by applying each of the potential solutions, in order to choose the most convenient for a specific situation.

2) Prediction of Required CPU Resources: In addition, the optimal amount of resources needed by the 5GC to perform the corresponding procedures under specific conditions has been proposed. This model provides information about the most cost-effective solution based on the input parameters (Packet rate, Packet size, Number of UE) and Throughput. In this way, it could help the network to choose whether an increase in scale is sufficient to avoid congestion or alternatively a migration is justified even at a higher cost. To this end, four representative algorithms from the related literature have been compared for the application of regressions to find the best fitting option. Hence, the following algorithms have been considered: Polynomial Regression (PR), KNN-based regression, RF-based regression and the SVM method [38].

Also, three commonly used Figures of Merit [39], [40], also called FOMs, have been taken into account to evaluate the proposed models:

- *Coefficient of determination* (R^2)*:* this FOM measures how close the data are to the fitted regression line. In general, the closer to 1 is R^2, the better the model fits the data (6).
- *Mean Absolute Error (MAE):* this metric evaluates the model by averaging the differences between the predicted and the target results. In this FOM, all the individual differences are weighted equally, so outlying errors will not penalize the results as much as the Mean Square Error metric. Still, it gives no information about how good is the prediction because it provides an absolute value (7).
- *Mean Absolute Scaled Error (MASE):* to give relative performance measure several FOMs can be used. As previously evaluated in related works, MAPE normalizes the error to Y_{Target} values, which brings some problems when dividing by zero. To overcome this, MASE is proposed for the present evaluation. MASE normalizes the obtained error to the mean error obtained by a naïve forecasting, which determines that the Y_i is the same as Y_{i-1}, without considering any other information (8).

$$R^2 = 1 - \frac{\sum_0^{N-1}\left(y_{i_{Target}} - \overline{Y_{Target}}\right)^2}{\sum_0^{N-1}\left(y_{i_{Predicted}} - \overline{Y_{Predicted}}\right)^2} \tag{6}$$

$$MAE = \frac{1}{n}\sum_0^{N-1}\left|y_{i_{Target}} - y_{i_{Predicted}}\right| \tag{7}$$

$$MASE = \frac{MAE}{\frac{1}{N-1}\sum_1^{N-1}\left|y_{i_{Train}} - y_{i-1_{Train}}\right|} \tag{8}$$

The implemented models and chosen FOMs are presented in Table V. On the one hand, KNN model obtains a better performance for almost all the proposed metrics, while SVMs tend to obtain a high MASE. On the other, MASE values higher than 1 determine that a naïve forecast would obtain a better performance than the ML model, although the model showed a good fit in terms of coefficient of determination. The forecasts for every metric show a good fit, although the MASE exceeds the unit value, which is assumed to be due to the high correlation between the outcome at the time instant and the next. For example, the network throughput is not expected to drop in a short period of time, since a constant throughput is set as the network input. Therefore, a naïve forecast will perform better than an ML model for the simplified scenario presented, with no variations in transmission rate or in the characteristics of the services deployed. Hence, a naïve model will not perform as well in a more complex scenario, so it would be necessary to make use of ML models to get a better estimate.

As presented above, predicting throughput, latency, packet loss and the needed resources for achieving an optimal performance is considered one of the main inputs to the NN management process in the 5GC scenario, what is obtained from the presented ML models, and what has been considered crucial in order to set an efficient decision-making process in trying to determine the best possible solution, while evaluating the impact it entails for the rest of the infrastructure.

Thus, the presented algorithms show the capability to infer network KPIs, supporting the task of managing a virtual infrastructure and handling Noisy Neighbours in a 5G virtualized network, what still remains a relevant challenge.

VI. CONCLUSION

In this work, the Noisy Neighbour (NN) problem has been presented in a virtualized 5G network environment. To analyze this matter, a complete 5G network framework has been implemented, including the execution of the core in a virtualization platform isolated from other network elements. On top of this virtualized entity, a novel Noisy Neighbour simulator specifically designed to perform a study on the impact and management of this issue, has been developed. The proposed NN simulator is able to take full effective control of the resources, thus abstracting from the underlying virtualization infrastructure.

Afterwards, the DoE methodology applied to the measurement campaign has allowed verifying the influence of the 5GC input parameters on the amount of demanded computing resources. In this sense, the high correlation obtained between packet rate and resource demand and the comparison of the different simulations has allowed inferring a proportionality relationship between both variables. On the contrary, the rest of the input factors (packet size and number of UEs) are significantly less correlated with 5GC resource consumption. The impact of the NN on 5G network performance has also been demonstrated. In this sense, the lack of resources caused by the emergence of this entity induces a drastic increase in latency and packet loss metrics while decreasing network throughput.

Having determined both the influence of the input parameters and the impact of the NN, a flow for the VNF life cycle management adapted to the 5GC environment has been proposed. Accordingly, two functions supported by ML algorithms have been determined as necessary. Namely, the NN identification task and the prediction of network performance indicators and resources required to maintain a load level.

In this way, the performance of the SVM model for the identification of the NN has been compared with the models used in the literature, where a more significant accuracy has been obtained with respect to previous works, reaching more than 99% accuracy. Furthermore, different variants of the model have been compared to evaluate the increase in the achieved accuracy as more network KPIs were added. As a result, the E2E delay of the packets transmitted by the UEs leads to a significant improvement in the accuracy of the model with respect to the addition of the rest of the metrics. Consequently, the importance of recording the E2E delay for the best performance of the NN identification algorithms is highlighted.

Besides, the prediction model obtained for the network metrics reaches coefficients of determination close to one, with KNN being the algorithm that shows the best fit. As a result, network KPIs inference is presented as a key input for the NN handling task, enabling the possibility of evaluating different potential scenarios for 5GC's VNFs.

As future work, a further test campaign is planned to extract more data for the different working contexts of the 5GC. This campaign will allow characterizing its behaviour more accurately. Also, early identification of the NN based on unsupervised methods is foreseen.

Moreover, the study of scenarios in which the 5GC functions are disaggregated among several virtual machines or containers is proposed, thus determining the impact of the NN on each of these functions and the performance of the network service. The influence of a defective assignment in a VNF (e.g., Access and Mobility Management Function AMF) on the other functions (e.g., UPF) depending on the load level could also be further analyzed.

Additionally, the validation of the results in a real virtualized 5G network from different vendors is suggested as one of the challenges for the enhancement of the present work, being necessary to contrast the results already obtained in the emulated environment.

References

[1] K. Abbas, M. Afaq, T. A. Khan, A. Rafiq, and W.-C. Song, "Slicing the core network and radio access network domains through intent-based networking for 5G networks," *Electronics*, vol. 9, no. 10, p. 1710, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/10/1710

[2] H. Basilier, J. Lemark, A. Centonza, and T. Asberg. "Applied network slicing scenarios in 5G." 2021. [Online]. Available: https://www.ericsson.com/491069/assets/local/reports-papers/ericsson-technology-review/docs/2021/applied-network-slicing-scenarios-in-5g.pdf

[3] *Study on Key Quality Indicators (KQIs) for Service Experience*, 3GPP Standard TS 32.862, Mar. 2016. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/32_series/32.862/32862-e00.zip

[4] C. Baena, S. Fortes, E. Baena, and R. Barco, "Estimation of video streaming KQIs for radio access negotiation in network slicing scenarios," *IEEE Commun. Lett.*, vol. 24, no. 6, pp. 1304–1307, Jun. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9031300/

[5] B. Blanco et al., "Technology pillars in the architecture of future 5G mobile networks: NFV, MEC and SDN," *Comput. Stand. Interfaces*, vol. 54, pp. 216–228, Nov. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920548916302446

[6] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017.

[7] S. Akundi, S. Prabhu, N. U. B. K, and S. C. Mondal, "Suppressing noisy neighbours in 5G networks: An end-to-end NFV-based framework to detect and suppress noisy neighbours," in *Proc. 21st Int. Conf. Distrib. Comput. Netw.*, New York, NY, USA, 2020, pp. 1–6. [Online]. Available: https://doi.org/10.1145/3369740.3372768

[8] R. Botez, J. Costa-Requena, I.-A. Ivanciu, V. Strautiu, and V. Dobrota, "SDN-based network slicing mechanism for a scalable 4G/5G core network: A Kubernetes approach," *Sensors*, vol. 21, no. 11, p. 3773, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/11/3773

[9] W. da Silva Coelho, A. Benhamiche, N. Perrot, and S. Secci, "Function splitting, isolation, and placement trade-offs in network slicing," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 2, pp. 1920–1936, Jun. 2022.

[10] A. A. Khan, M. Abolhasan, W. Ni, J. Lipman, and A. Jamalipour, "An end-to-end (E2E) network slicing framework for 5G vehicular ad-hoc networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 7, pp. 7103–7112, Jul. 2021.

[11] T. Irshad, "Design and implementation of a testbed for network slicing," M.S. thesis, Dept. School Electr. Eng., Aalto Univ., Espoo, Finland, 2018. [Online]. Available: http://urn.fi/URN:NBN:fi:aalto-201802231619

[12] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, 2017, pp. 127–140. [Online]. Available: https://doi.org/10.1145/3117811.3117831

[13] U. Margolin, A. Mozo, B. Ordozgoiti, D. Raz, E. Rosensweig, and I. Segall, "Using machine learning to detect noisy neighbors in 5G networks," 2016, *arXiv:1610.07419*.

[14] C. Jo, Y. Cho, and B. Egger, "A machine learning approach to live migration modeling," in *Proc. Symp. Cloud Comput.*, New York, NY, USA, 2017, pp. 351–364. [Online]. Available: https://doi.org/10.1145/3127479.3129262

[15] M. Ersue, "ETSI NFV management and orchestration-An overview," IETF, Vancouver, BC, Canada, document IETF #88, 2013.

[16] M. Abu-Lebdeh, D. Naboulsi, R. Glitho, and C. W. Tchouati, "On the placement of VNF managers in large-scale and distributed NFV systems," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 875–889, Dec. 2017.

[17] S. Lange et al., "A network intelligence architecture for efficient VNF lifecycle management," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 1476–1490, Jun. 2021.

[18] Y. Anser, J.-L. Grimault, S. Bouzefrane, and C. Gaber, "Energy-aware service level agreements in 5G NFV architecture," in *Proc. 8th Int. Conf. Future Internet Things Cloud (FiCloud)*, 2021, pp. 377–382.

[19] T. V. K. Buyakar, A. K. Rangisetti, A. A. Franklin, and B. R. Tamma, "Auto scaling of data plane VNFs in 5G networks," in *Proc. 13th Int. Conf. Netw. Service Manag. (CNSM)*, 2017, pp. 1–4.

[20] C. H. T. Arteaga, A. Ordoñez, and O. M. C. Rendon, "Scalability and performance analysis in 5G core network slicing," *IEEE Access*, vol. 8, pp. 142086–142100, 2020.

[21] S. Van Rossem, W. Tavernier, D. Colle, M. Pickavet, and P. Demeester, "Automated monitoring and detection of resource-limited NFV-based services," in *Proc. IEEE Conf. Netw. Softw. (NetSoft)*, 2017, pp. 1–5.

[22] H. Bouattour, Y. B. Slimen, M. Mechteri, and H. Biallach, "Root cause analysis of noisy neighbors in a virtualized infrastructure," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2020, pp. 1–6.

[23] A. Cárdenas and D. Fernández, "Network slice lifecycle management model for NFV-based 5G virtual mobile network operators," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, 2020, pp. 120–125.

[24] I. Sarrigiannis, K. Ramantas, E. Kartsakli, P.-V. Mekikis, A. Antonopoulos, and C. Verikoukis, "Online VNF lifecycle management in an MEC-enabled 5G IoT architecture," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4183–4194, May 2020.

[25] M. Moradi, M. Ahmadi, and R. Nikbazm, "Comparison of machine learning techniques for VNF resource requirements prediction in NFV," *J. Netw. Syst. Manag.*, vol. 30, no. 1, pp. 1–29, 2022.

[26] B. Ordozgoiti, A. Mozo, S. G. Canaval, U. Margolin, E. J. Rosensweig, and I. Segall, "Deep convolutional neural networks for detecting noisy neighbours in cloud infrastructure," in *Proc. ESANN*, 2017, pp. 571–576.

[27] P. Veitch, E. Curley, and T. Kantecki, "Performance evaluation of cache allocation technology for NFV noisy neighbor mitigation," in *Proc. IEEE Conf. Netw. Softw. (NetSoft)*, 2017, pp. 1–5.

[28] "LoadCore 5G core testing." Keysight Technologies. Jan. 2021. [Online]. Available: https://www.keysight.com/es/en/assets/3120-1180/data-sheets/LoadCore-5G-Core-Testing.pdf

[29] L. Tomás, C. Vázquez, J. Tordsson, and G. Moreno, "Reducing noisy-neighbor impact with a fuzzy affinity-aware scheduler," in *Proc. Int. Conf. Cloud Autonomic Comput.*, 2015, pp. 33–44.

[30] T. Lorido-Botrán, S. Huerta, L. Tomás, J. Tordsson, and B. Sanz, "An unsupervised approach to online noisy-neighbor detection in cloud data centers," *Expert Syst. Appl.*, vol. 89, pp. 188–204, Dec. 2017.

[31] J. Al Sadi, Y. L. Nimir, and L. Ferreira, "Experimental optimization: 3 level full factorial design and variation of processing parameters for polycarbonate colors," in *Proc. Adv. Sci. Eng. Technol. Int. Conf. (ASET)*, 2018, pp. 1–5.

[32] G. M. Yilma, Z. F. Yousaf, V. Sciancalepore, and X. Costa-Perez, "Benchmarking open source NFV MANO systems: OSM and ONAP," *Comput. Commun.*, vol. 161, pp. 86–98, Sep. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366420305946

[33] C. A. Ul Hassan, M. S. Khan, and M. A. Shah, "Comparison of machine learning algorithms in data classification," in *Proc. 24th Int. Conf. Autom. Comput. (ICAC)*, 2018, pp. 1–6.

[34] B. Pavlyshenko, "Using stacking approaches for machine learning models," in *Proc. IEEE 2nd Int. Conf. Data Stream Min. Process. (DSMP)*, 2018, pp. 255–258.

[35] S. Fortes, C. Baena, J. Villegas, E. Baena, M. Z. Asghar, and R. Barco, "Location-awareness for failure management in cellular networks: An integrated approach," *Sensors*, vol. 21, no. 4, p. 1501, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/4/1501

[36] D. Palacios, S. Fortes, I. De-La Bandera, and R. Barco, "Self-healing framework for next-generation networks through dimensionality reduction," *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 170–176, Jul. 2018.

[37] A. Herrera-Garcia, S. Fortes, E. Baena, J. Mendoza, C. Baena, and R. Barco, "Modeling of key quality indicators for end-to-end network management: Preparing for 5G," *IEEE Veh. Technol. Mag.*, vol. 14, no. 4, pp. 76–84, Dec. 2019.

[38] E. Y. Boateng, J. Otoo, and D. Abaye, "Basic tenets of classification algorithms K-nearest-neighbor, support vector machine, random forest and neural network: A review," *J. Data Anal. Inf. Process.*, vol. 8, pp. 341–357, Jan. 2020.

[39] B. Vega-Marquez, C. Rubio-Escudero, I. A. Nepomuceno-Chamorro, and A. Arcos-Vargas, "Use of deep learning architectures for day-ahead electricity price forecasting over different time periods in the Spanish electricity market," *Appl. Sci.*, vol. 11, no. 13, p. 6097, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/13/6097

[40] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLoS One*, vol. 13, Mar. 2018, Art no. e0194889.

Francisco Muro received the degree in telematics engineering from the University of Málaga, Spain, in 2021, where he works as a Research Assistant, focusing on performance monitoring and optimization in virtualized wireless networks.

Eduardo Baena received the M.Sc. degree in telecommunication engineering from the Universidad de Granada, Spain, in 2010. He is currently pursuing the Ph.D. degree with the University of Málaga, Spain, focus on unlicensed band applications and advanced network management schemes. He has held various industry positions in several companies, including operators, service providers, and manufacturers. Since 2017, he has been working as a Lecturer and a Researcher with the Department of Communications Engineering, University of Málaga, Spain.

Sergio Fortes (Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of Málaga, Spain. He began his career in the field of satellite communications, holding positions in European Space Agencies, where he participated in various research and consultant activities on broadband and aeronautical satellite communications. In 2012, he joined the University of Málaga, where his research is focused on self-organizing networks for cellular communications.

Lars Nielsen received the Master of Science degree in networks and distributed systems and the Ph.D. degree from AAU in 2013 and 2017, respectively. He works as a R&D Engineer at Keysight participating in various international projects as H2020 VINNI.

Raquel Barco received the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of Málaga, Spain. In 2000, she joined the University of Málaga, where she is currently a Full Professor. She has worked in projects with major mobile communications operators and vendors and is an author of more than 100 high-impact papers.