BUILD YOUR OWN CLOSED LOOP: GRAPH-BASED PROOF OF CONCEPT IN CLOSED LOOP FOR AUTONOMOUS NETWORKS

Jaime Fúster de la Fuente[1], Álvaro Pendás Recondo[2], Paul Harvey[3], Tarek Mohamed [4], Chandan Singh[5], Vipul Sanap[5], Ayush Kumar[5], Sathish Venkateswaran[6], Sarvasuddi Balaganesh[6], Rajat Duggal[6], Sree Ganesh Lalitaditya Divakarla[7], Vaibhava Krishna Devulapali[7], Ebeledike Frank Chukwubuikem[8], Emmanuel Othniel Eggah[8], Abel Oche Moses[8], Nuhu Kontagora Bello[8], James Agajo[8], Wael Alron[9], Fathi Abdeldayem[9], Melanie Espinoza Hernández[10], Abigail Morales Retana[10], Jackeline García Alvarado[10], Nicolle Gamboa Mena[10], Juliana Morales Alvarado[10], Ericka Pérez Chinchilla[10], Amanda Calderón Campos[10], Derek Rodríguez Villalobos[10], Oscar Castillo Brenes[10], Kodandram Ranganath[6], Ayushi Khandal[6], Rakshesh P Bhatt[6], Kunal Mahajan[11], Prikshit CS [11], Ashok Kamaraj[6], Srinwaynti Samaddar[6], Sivaramakrishnan Swaminathan[6], M Sri Bhuvan[12], Nagaswaroop S N[12], Blessed Guda[13], Ibrahim Aliyu[14], Kim Jinsul[14], Vishnu Ram[15]

[1]Rakuten Mobile, [2]University of Oviedo, [3]University of Glasgow, [4]Dept of Electronics and Electrical Communications, Fayoum University, Egypt , [5]TCS, India, [6]Nokia Networks Bengaluru, India, [7]PES College, India, Bengaluru, [8]Dept. of Computer Engineering, Federal University of Technology, Minna, [9]du, Dubai, [10]Descubre Robótica, Costa Rica, [11]RVCE, Bengaluru, India, [12]DSCE, Bengaluru, India, [13]Dept. of Electrical and Computer Engineering/AI, Carnegie Mellon University, Africa, [14]Dept. of ICT Convergence System Engineering, Chonnam National University, Gwangju, South Korea, [15]Independent Expert

NOTE: Corresponding author: Ibrahim Aliyu, ibal2010@yahoo.com

Abstract *– Next Generation Networks (NGNs) are expected to handle heterogeneous technologies, services, verticals and devices of increasing complexity. It is essential to fathom an innovative approach to automatically and efficiently manage NGNs to deliver an adequate end-to-end Quality of Experience (QoE) while reducing operational expenses. An Autonomous Network (AN) using a closed loop can self-monitor, self-evaluate and self-heal, making it a potential solution for managing the NGN dynamically. This study describes the major results of building a closed-loop Proof of Concept (PoC) for various AN use cases organized by the International Telecommunication Union Focus Group on Autonomous Networks (ITU FG-AN). The scope of this PoC includes the representation of closed-loop use cases in a graph format, the development of evolution/exploration mechanisms to create new closed loops based on the graph representations, and the implementation of a reference orchestrator to demonstrate the parsing and validation of the closed loops. The main conclusions and future directions are summarized here, including observations and limitations of the PoC.*

Keywords – AI/ML, anomaly, autonomous networks, closed loop, graph, KPI, PoC, use case

1. INTRODUCTION

Next Generation Networks (NGNs) will face network management bottlenecks from a wide range of services such as online gaming, AR/VR, intelligent transport systems, smart cities, metaverse etc. They are expected to cope with heterogeneous technologies, services, verticals and increasingly complex devices [1-3]. Thus, it is essential to derive an innovative approach to automatically and efficiently manage NGNs to deliver adequate end-to-end Quality of Experience (QoE) while reducing operational costs.

An Autonomous Network (AN) where a network can self-monitor, self-evaluate and self-heal is a potential solution to managing the NGN. Closed-loop automation solutions support autonomous behaviour through a function that monitors the network and uses feedback signals to achieve the desired state. The AN is an important part of the solution in all areas of network management, including planning, audit, inventory, optimization, security, orchestration and quality of experience. [4].

The International Telecommunication Union Focus Group on Autonomous Networks (ITU FG-AN) organized a build-a-thon challenge in 2023, in the form of a PoC project to create a crowdsourced baseline representation for AN closed loops (controllers), review and analyse them, and publish them in an open repository. Additionally, this effort creates reference demonstration of supporting tools ("AN orchestrator", "openCN" [4], evolution controller [FGAN-I-198]) and triggers technical discussions on the standard format for representing closed loops (controllers). Further extensions on

top of the baseline and reference implementations are possible. The PoCs created are majorly based on the use case described in [4] and are listed below and discussed in detail in subsequent sections.

The main problems addressed by the use case covers a solution to support the operation of closed-loop controllers. In contrast, other use cases develop a closed-loop controller for a specific problem. The use case that offers a solution to support closed loop operation includes a knowledge import and export system (use case 1); secure and traceable evolution system for AN (use case 2); link prediction for inferring new relations between AN use case actors (use case 3); monitoring of the most significant KPIs and the automatic detection of anomalies in 5G and beyond (use case 5); and network resource allocation for emergency management based on a closed loop (use case 7). The other use cases that develop a closed-loop controller include slip detection (and force estimation) and object detection in a robotic grasping application (use case 4) and autonomous agents (with varied competence) design in networks (use case 6).

A graph-based representation of use cases is uniformly applied to all the AN use cases. The main contributions of this study are summarised as follows:

- **Use case 1 – Knowledge import and export system:** This study implemented a PoC for the autonomous evolution of controllers based on the FGAN architecture [4]. Additionally, a PoC is described which includes the subsequent deployment of any controller obtained from the evolution. Topology and Orchestration Specification for Cloud Applications (TOSCA) representation of the chosen controller is included as part of this study.
- **Use case 2 – Secure and traceable evolution system for AN:** To achieve this level of trust between disparate functions in the network, a proper decentralization of the evolution process needs to be ensured. We specifically address important issues, such as the process's security, auditability, ease of understanding, explainability and scalability using Ethereum and the Interplanetary File System (IPFS).
- **Use case 3 – Link prediction for inferring new relations between AN use case actors:** We explore the link prediction algorithm to investigate and derive new use cases for autonomous networks to understand future use cases. The algorithm attained 86% prediction accuracy and identified more than 6.5k possible relations that can result in new use cases and can be studied in future.
- **Use case 4 – Slip detection (and force estimation) and object detection in a robotic grasping application:** A low latency closed loop system is developed between the haptic hands/algorithms and real robotic hand (Allegro Hand) using a MEC testbed. In addition, a Long Short-Term Memory (LSTM) model and Random Forest (RF) classifier for slip detection (and force estimation) and object detection were developed, tested and validated.
- **Use case 5 – Monitoring of the most significant KPIs and the automatic detection of anomalies in 5G and beyond:** An unsupervised deep-learning model, LSTM-based Variational Autoencoder (LSTM-VAE), is proposed to detect and forecast the most significant KPI's anomalies to better network troubleshooting. The model validation and evaluation is done using Mean Absolute Error (MAE) and Reconstruction Error (RE).
- **Use case 6 – Design of autonomous agents (with varied competence) in networks:** Data is collected from sensors in the network, such as logs, packet measurements, deep packet inspection, etc. and is managed by autonomous agents with minimal human intervention.
- **Use case 7 – Network resource allocation for emergency management based on closed loop:** All incoming KPIs and slice configurations are monitored continuously, including any changes in slice configurations. New slice deployments are analysed, and an RL agent optimises the slice configurations accordingly using calculated rewards based on SLAs. Also, a machine learning-based closed loop for log anomaly detection, prediction and probable root cause analysis is investigated.
- Lastly, the paper presents business and design considerations for adopting ANs. A reference design is also produced to this effect on the AN-enabled end-to-end supply chain.

This paper is divided into the following sections: Section 1.1 discusses the background and key concepts of the study. Section 2 presents use case 1 on the knowledge import and export system, and Section 3 presents the evolution and blockchain:

an autonomous network architecture PoC (use case 2). New use cases based on a link prediction algorithm are presented (use case 3) in Section 4. Meanwhile, Section 5 presents a low latency closed loop for robotic grasping as use case 4. Section 6 describes use case 5 on monitoring the most significant KPIs and automatically detecting anomalies. Sections 7, 8 and 9 present use cases on autonomous agents in networks (use case 6), network resource allocation for AN management (use case 7) and autonomous log troubleshooting, respectively (use case 8). Lastly, Section 10 concludes the paper and provides future direction.

1.1 Background

This section presents some background on the technologies and key concepts used in this study. The use cases and architecture for autonomous networks are described in [4] and [5], respectively. Fig. 1 explains the architecture for autonomous networks [5], which forms the basis of studies in this paper.

As explained in [5] the main concepts behind autonomous networks, which are elaborated here, are exploratory evolution, real-time responsive online experimentation and dynamic adaptation. To study and analyse use cases along these concepts in networks, a basic building block called “controller” is introduced. A controller can be described as an open/closed loop workflow composed of several integrated modules which can be developed independently to solve a specific problem or enforce a given requirement. Controllers are used in the use cases to further elaborate autonomous networks and the key concepts required to enable them.

The concept of exploratory evolution introduces the mechanisms and processes of exploration and evolution to adapt a controller in response to changes in the underlay network. These processes generate new controllers or update (evolve) existing controllers to respond to such changes and solve the situation or task at hand more appropriately.

The continuous process, based on monitoring and optimization of deployed controllers in the underlay network, is called real-time responsive online experimentation. Meanwhile, dynamic adaptation is the final concept in equipping the network with autonomy and the ability to handle new and hitherto unseen changes in network scenarios.

With consideration of the above concepts, an autonomous network is a network which can generate, adapt and integrate controllers at run time using network-specific information and can realize exploratory evolution, real-time responsive online experimentation and dynamic adaptation.

The rest of the paper describes each use case and the corresponding design along with the PoC implementations.

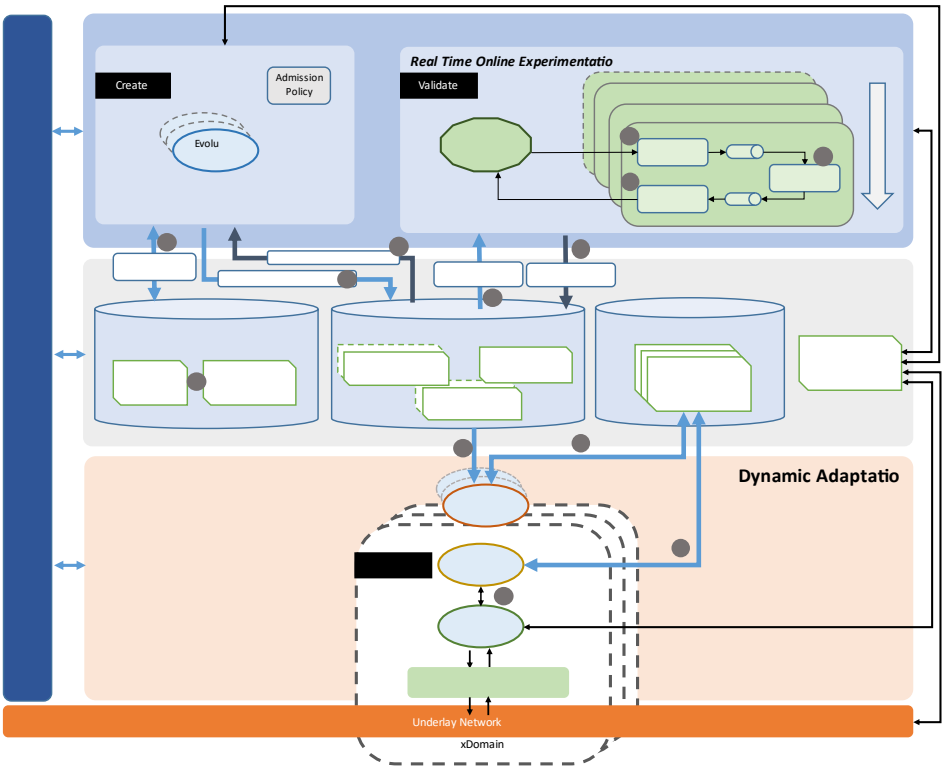


Fig. 1 – High-level framework for evolutionary-driven autonomous network [5]

2. USE CASE 1: KNOWLEDGE IMPORT AND EXPORT SYSTEM

2.1 The use case design

Knowledge, and its exchanges between AN components, is the deriving component behind the main concepts of autonomous networks-exploratory evolution, real-time responsive online experimentation and dynamic adaptation-elaborated in Section 1.1. The knowledge involves the collection and management of resources that help the AN to perform its function. Knowledge in ANs ranges from autonomous system environmental data representation, actions/consequences and key configuration options to potential parameter indices. It includes metadata, AN component status, structured and unstructured data derived from different sources and actors etc. Since the controllers are workflows which involve knowledge exchanges, we consider the knowledge import and export system use case.

To this end, we describe a basic concept that relates to knowledge import and export. According to [5], a module is "a building block consisting of executable code and a module specification from which controllers are assembled". A module's specification includes its input and output interfaces and a metadata description of its functionality. Examples include aggregation functions, DNS configuration interfaces, an entire Deep Neural Network (DNN) model, a single layer of a DNN model etc. On the other hand, a controller can be considered as a software closed loop composed of modules. A controller instance is "an executable representation of a controller including modules, their configurations, and parameter values".

In this use case, we examine the knowledge import and export system to fulfil the key concepts of ANs (evolution, experimentation and adaptation) while minimizing human intervention. The system consists of an AN orchestrator, Knowledge Base (KB) manager, auto-controller, baseline repository of intents for different forms of controllers called open CN, Machine Learning (ML) pipeline and human operator.

The *orchestrator* is responsible for managing workflows and processes in the AN and steps in the lifecycle of controllers. To manage the workflows and processes, the AN orchestrator coordinates various other functions in and outside the AN. Being part of the management plane, the AN orchestrator provides an interface to human operators in the form of reports regarding the functioning of the AN and, where applicable, human interfaces for configuring the AN.

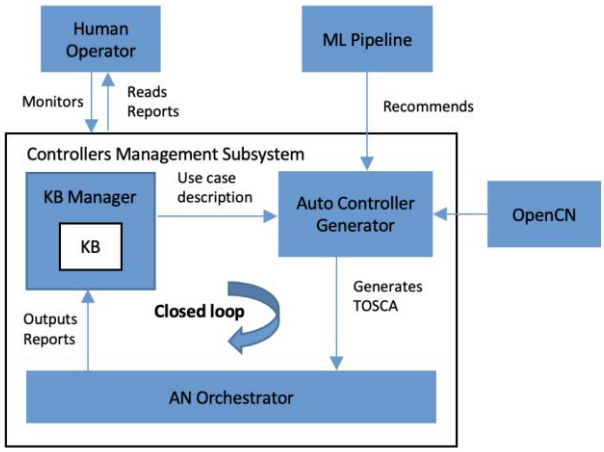


Fig. 2 – High-level flow chart for the creation and parsing of example intent

The *Knowledge Base manager* is a subsystem, part of the controller management subsystem in Fig. 1, that manages storage, querying, export, import and optimization, and updates knowledge derived from different sources, including structured or unstructured data from various components or other subsystems. KB manager is a node that optimizes and manages data available on the closed loop but requires a node that can host its resources. Since the AN orchestrator, part of the controller management subsystem in Fig. 2, can host node resources, it is designed to be a host to the KB manager. Knowledge in the AN is a collection of resources that helps solve a specific problem. A knowledge base component manages knowledge derived from and used in ANs. It is updated and accessed by various components in the AN. The knowledge includes metadata derived from the capabilities and status of AN components; it can be derived from different sources, including structured or unstructured data. This knowledge is stored and exchanged as part of interactions of the AN components with the knowledge base.

The *auto-controller generator,* part of the controller management subsystem in Fig. 2, represents a generic software component that can be managed and run by a TOSCA compute node type. It generates controller specifications using the existing repository in OpenCN, which is an open repository for storing controllers, shown as a knowledge base in Fig.1 and an analytics function aided by AI/ML. Meanwhile, the Open CN stores the controllers for the AN.

The *ML pipeline* node, which is part of the underlay network in Fig. 2, can learn and adapt without following explicit instructions, using algorithms and statistical models to analyse and draw inferences from patterns in data. It hosts analytics and recommends controllers in the AN. On the other hand, the *human operator,* interfacing via the controller management subsystem in Fig. 2, is just like a user-friendly interface for human base instructions.

2.2 Use case PoC implementation

The TOSCA metamodel uses the concept of service templates that describe cloud workloads as a topology template, which is a graph of node templates modelling the components a workload is made up of and of relationship templates modelling the relations between those components. TOSCA service templates are instantiated at runtime using a TOSCA orchestrator (xOpera), and the order of component instantiation is based on the relationship between components. TOSCA is one of the best and most often used automated testing tools and is widely employed in large-scale applications to achieve successful outcomes.

In the implementation, the YAML file is generated and stored in a graph database using neo4j. After the basic YAML template of the use case was written, Ruamel (a Python YAML editing library) was used to populate the nodes of the use case with the actors and relationships from our use case in TOSCA by using information obtained from querying the graph database. Finally, the generated YAML was then validated by parsing it using xOpera. Fig. 3. shows the flow chart for service template generation from the graphDB, while Fig. 4 shows the corresponding neo4j graph.

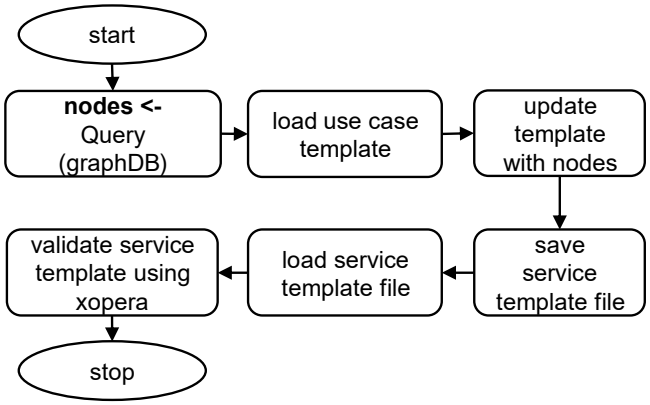


Fig. 3 – Flow chart of the service template generation from the graphDB

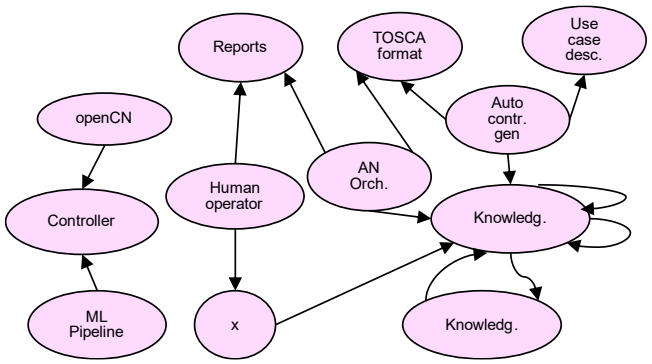


Fig. 4 – Import and export of knowledge neo4j graph

3. USE CASE 2: EVOLUTION AND BLOCKCHAIN: AN AUTONOMOUS NETWORK ARCHITECTURE POC

3.1 Use case design

In order to warrant traceability and security in the evolution process, every function in the system has to be able to trust and verify unequivocally the authenticity of each evolved controller. To achieve this level of trust between disparate functions in the network, a proper decentralization of the evolution process needs to be ensured. Ethereum and the Interplanetary File System (IPFS) are used to solve this problem. Trust, authentication and traceability are ensured thanks to Ethereum, a Distributed Ledger Technology (DLT) with Smart Contract (SC) functionality, while integrity, scalability and fault tolerance are achieved with the IPFS, a distributed and decentralized file system that implements content addressing.

In an AN, the different nodes from which it is composed may be capable of making localized decisions without depending on a central authority for validation. Consequently, realising autonomy for such a set of independent nodes requires trust. Therefore, for evolution to take place in a fully decentralized system in a trustworthy manner, the evolution process needs to be auditable and participating nodes have to be authenticated.

Being a blockchain, Ethereum has the capability to record and trace transactions, which provides traceability and trust throughout the process. Thanks to the use of SCs, it also has the ability to implement Decentralized Applications (DApps) that enable more complex processes to take place in the blockchain.

Even though the Ethereum blockchain provides a means to ensure decentralization and audibility throughout the evolution process, the nature of DLTs makes it unsuitable for storing large amounts of data or executing complex applications. A distributed means of storage is therefore required to safeguard a decentralized way of storing artefacts, without sacrificing scalability or fault tolerance.

The Interplanetary File System (IPFS) is a peer-to-peer hypermedia protocol for storing and sharing data in a distributed file system. Put in simple terms, the IPFS allows for sharing and storing data of any kind (files, websites, applications) in a distributed and decentralized fashion, without depending on a single end point or source's availability to access them.

This section presents a PoC of the decentralized controller evolution architecture for ANs using a distributed marketplace based on the FGAN architecture (FGAN-I-198 [5]). Such a marketplace serves as a decentralized mechanism for the secure and traceable evolution of controllers, which is achieved by a private Ethereum blockchain and a distributed and decentralized file system, the IPFS.

We specifically address important issues, such as the process's security, auditability, explainability and scalability. Additionally, the PoC also includes the subsequent automatic deployment of any controller obtained from the evolution. This last part contains the TOSCA parsing of a REST service for the chosen controller.

3.2 Use case PoC implementation

Fig. 5 shows the high-level Neo4J graph representation for the PoC closed loop, which has in its scope all of the concepts presented in Fig. 1, except for the selection controller and operation controller. Note that the use of a Digital Twin (DT) is considered for real-time online experimentation. A DT, which is a concept described in [6], is "a mathematical representation of a physical and/or logical object". For example, a DT could be a Graph Neural Network (GNN) that reproduces the behaviour of a network, producing outputs such as latency or packet loss rate, given inputs that describe its state and policy. This PoC uses the concept of a DT in the architecture but does not focus on its implementation, which is a problem beyond the scope of this research.

The blue nodes in Fig. 5 are labelled as actors in the graph, while the green ones are denoted as elements. Actors are controllers that can evolve, although the evolution of actors is beyond the scope of this PoC. The element includes supporting artefacts, such as software modules, evolvable

controllers, repositories etc. The evolution controller (Evol_Ctr) stores the list of available modules and uses them to compose an evolvable controller (Evola_Ctr) that is sent to the marketplace (Mark_Plc). Although the marketplace is represented as a single node, it is a distributed network of several nodes that are part of the rest of the actors. That means that the marketplace is distributed across all the instantiations of the evolution controller, experimentation manager, digital twin and curation controller, ensuring the security and traceability of the evolution process. In this PoC, both modules and controllers are implemented as Python classes. The source code for both definitions is available in [7], in the *Mod_Ctr.py* file, which is reproduced in all the nodes. That all nodes work with the same definition is among the important assumptions. The important assumptions are as follows:

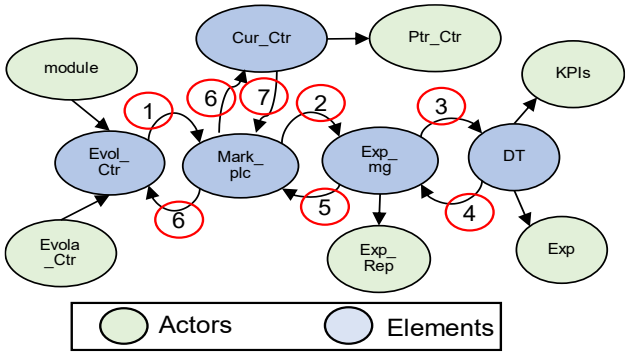


Fig. 5 – Neo4J graph representation of the high-level architecture of the PoC

- The list of available modules and the definition of the classes Module and Controller are contained in the *Mod_Ctr.py* file, with each node (blue circles in Fig. 5) having an identical copy of it. In the case of introducing new modules, something that is not considered in this PoC, a protocol for distributing the information and ensuring consistency is required. However, the marketplace provides the perfect tool for implementing this feature.
- Since the PoC is focused on laying the groundwork for the evolution architecture, the modules and controller considered are rather simple, reproducing the behaviour of simple mathematical operations. However, we assume that this approach is sufficient for proving the feasibility of our architecture and that it can be improved in the future.
- As for the evolution, the algorithm uses a random combination of modules, avoiding repetition for which their result testing is known. We assume that our use of random module combinations is structurally equivalent in this context to those controllers produced by evolution. Of course, they are not expected to be equivalent in their operation.

All these assumptions simplified the PoC implementation without limiting its effectiveness when laying the groundwork for the evolution of controllers in a distributed system. New features can be added with a few or no changes in the base work presented here.

The class *Module* has four attributes: module id, function, parameter and representation.

The list of possible functions is limited to the mathematical operations add, subtract, pow and multiple. These functions are also defined as Python functions in the file *Mod_Ctr.py*.

Fig. 6 shows the Python code for the multiple functions, *f_mul*. Fig. 7 shows an instantiation of a *Module* object with *f_mul* as a function with float 3 as a parameter. The id is a string explanation of the function, while the representation is the symbol that denotes the mathematical operator. The *Module* class has the method *execute* represented in Fig. 8 that, given an input, returns the output of applying the function with the designed parameter. Once a *Module* object is instantiated it is possible to modify its parameter with the method *set_method.*

```
def f_mul (x, param):
    return x * param
```

Fig. 6 – Python code for the definition of the function *f_mul*

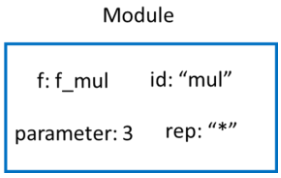


Fig. 7 – Representation of a *Module* object

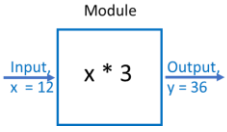


Fig. 8 – Representation of the method *execute* of a *Module* object

The class *Controller* has three attributes:

- the unique controller id
- a list of the modules that compose the controller
- a list of the parameters for each of its modules.

Fig. 9 represents the instantiation of a *Controller* object. In that case, the two modules implement the functions subtract and multiply tuned with parameters 2 and 10, respectively. After the object is instantiated, it is possible to call the method to *execute,* as represented in Fig. 10. The *executed* method for each *Module* object will be called in order, being the output of each phase and the input for the next. The controller also has a method (get_dict) that returns a Python dictionary describing its attributes.

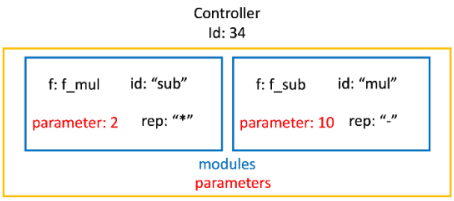


Fig. 9 – Representation of a *Controller* object

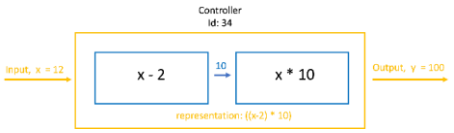


Fig. 10 – Representation of the method *execute* of the *Controller* class

A Python dictionary can be easily converted into a JavaScript Object Notation (JSON) file, the selected format for transmitting the controller information between nodes. The transmitter node sends the JSON description of a controller. Since the receiver also has the list of all available modules, it can recreate a copy of the controller by calling the function *json_to_ctr (dictionary)*. We have chosen JSON since it is a lightweight data-interchange format that is readable by humans. Fig. 11 shows the JSON representation of the *Controller* object introduced in Fig. 9.

3.3 Experimentation and evolution

All blue nodes in Fig. 5 communicate by exchanging JSON files via the Hypertext Transfer Protocol (HTTP). On the marketplace side, we have used the language JavaScript for handling the HTTP communication, while in the rest of the blue nodes, we have chosen the combination of the Python framework Flask [8] and Gunicorn [9], a Python HTTP server for Web Server Gateway Interface (WSGI) applications.

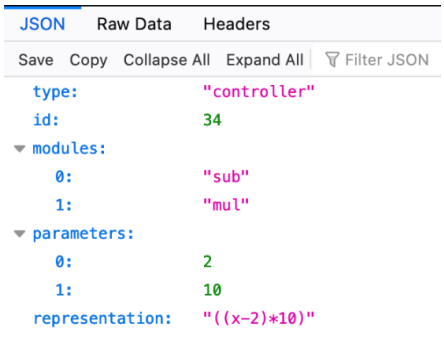


Fig. 11 – JSON representation of a *Controller* object extracted from the marketplace

Following the architecture presented in Fig. 5, the evolution controller (Evol_Ctr) sends the JSON representation of a new evolvable controller (Evola_Ctr) as the one shown in Fig. 11. After storing the JSON file, the marketplace forwards it to the experimentation manager (Exp_Mg). This node is configured with the DT by sending both the JSON representation of the controller and a list of experiments to be run. This PoC presents a simple case where only two experiments implemented as Python functions are available in the DT.

- ***Average*:** Return the average of the controller output after 100000 iterations, being the input drawn from a uniform distribution between 1 and 10.
- ***Value*:** Given a fixed input, return the absolute value of the difference between the controller output and a desired value. The input is 5 by default, and the desired output is 35.

The DT instantiates a controller object using the JSON representation and, after running the experiments, sends back to the experimentation manager a JSON containing the results or Key Performance Indicators (KPIs, Fig. 5). The experimentation manager composes an experiment report (Exp_Rep) that is a JSON containing the id of the controller under testing and its results in each experiment. The experiment report is sent to the marketplace and then forwarded to the curation controller (Cur_Ctr). An example of an experiment report (running the experiments with the default settings) is shown in Fig. 12. The report is also forwarded to the evolution controller since the results are useful feedback if an evolutionary algorithm is implemented.

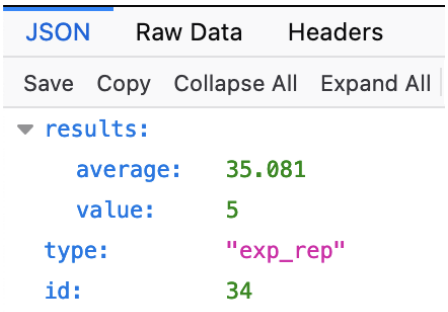


Fig. 12 – JSON representation of the experiment report of the controller described in Fig. 11, extracted from the marketplace

Based on the experiment report, the curation controller (Cur_Ctr) will decide if the evolvable controller referenced by its id is promoted to the protected controller (Ptr_Ctr) for one or more experiments. Treating the protected category independently for each experiment is based on the idea that a controller suitable for a specific use case may perform poorly if the case changes. For example, Fig. 13 shows the notification of the curation controller to the marketplace, informing that the controller with id 34 (Fig. 11) is a protected controller for the experiment/use case *value*. Note that the same controller does not have the category of protected for the experiment *average*. This decision depends on the criteria followed by the curation controller, which for this example is:

- ***Average*:** protected controller if the result is greater than 500.
- ***Value*:** protected controller if the result is smaller than 20.

Therefore, these values serve as configurable thresholds to decide if an evolvable controller may be promoted to a protected controller.

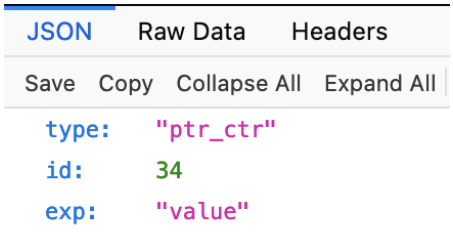


Fig. 13 – JSON file registration of the protected controller, extracted from the marketplace

Fig. 14 shows the functional level architecture of the PoC, where each node represents a specific technology implemented. As can be seen from the picture, each one of the actors shown in Fig. 5 is composed of a subset of services representing a specific function within the PoC. Different technologies are used depending on the type of functionality. A description of each type of service and the technology it uses is provided below:

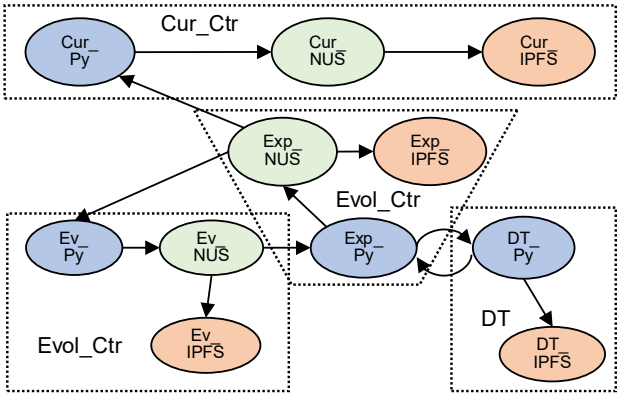


Fig. 14 – Functional level architecture of the PoC

- **Experimentation and evolution:** responsible for each actor's evolution and experimentation logic. Different actors have different functionalities, so implementation varies from actor to actor (the Cur_Ctr promotes controllers based on experimentation reports, the Evol_Ctr creates new controllers and sends them to the Exp_Mg etc.).
  - Programming language/frameworks: Python.
  - Naming convention: *actor name abbreviation + _Py* (e.g. *Cur_Py* for curation controller)
- **Marketplace:** services that make up the functionalities of the marketplace.
  - Ethereum service: responsible for receiving and executing transactions through SCs in Ethereum. These transactions include uploading a file to the IPFS and retrieving an uploaded file by its CID or by its id.
  - Programming languages/frameworks: Node.js, Hardhat, Solidity.
  - Naming convention: *actor name abbreviation + _NJS* (e.g. *Cur_NJS* for curation controller)
  - IPFS service: node connecting each actor to a private IPFS distributed file system network.
  - Naming convention: *actor name abbreviation + _IPFS* (e.g. *Cur_IPFS* for curation controller)

All independent services have been implemented through docker containers, orchestrated and configured via a docker compose *(docker-compose.yml)* file. Each container is interconnected, creating a network as shown in Fig. 15, from which it can be seen that each container/service of the system has an IP address and a specific set of ports exposed for communication with other nodes within the network. The distribution of these ports and addresses is detailed in Table 1.

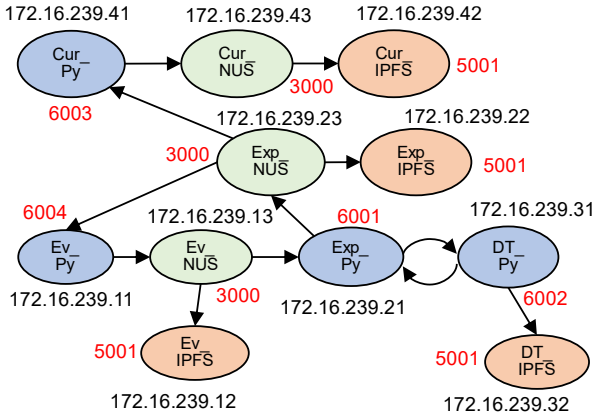


Fig. 15 – Network architecture of the PoC

3.4 The marketplace

The designed marketplace is a distributed network of several interconnected nodes rather than a single network node that makes up the PoC. As shown in the previous subsection, these nodes comprise docker containers incorporating two different services (Ethereum and IPFS) that make up the marketplace and are included in every actor that is part of the PoC, as shown in Fig. 5.

This means that the marketplace is distributed across all the instantiations of the evolution controller, experimentation manager, DT and curation controller, connecting every actor to the traceable ledger of evolution artefacts created throughout the process and stored in the IPFS and accessible through the private Ethereum blockchain. Fig. 16 presents an example of how the marketplace works when uploading an artefact. The process is described below:

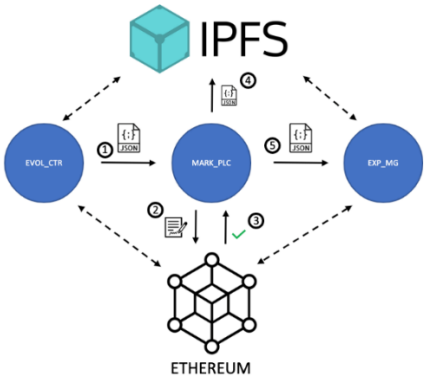


Fig. 16 – Example of uploading process in the marketplace

1. The evolution controller sends an evolvable controller in the form of a JSON file to the marketplace. This JSON file has the structure described in Section 3.1, containing information that identifies the controller according to its definition and is sent via a POST HTTP request to port 3000 of the *evol-ctr-mkp* container, which handles the HTTP API of the marketplace (see Table 1).
2. The marketplace receives the JSON artefact via port 3000 and registers the file into Ethereum via the *Marketplace.sol* SC. The marketplace runs the operation*.mjs* script, which is contained in every *_NJS* container, handles the marketplace operation, interconnecting Ethereum with the IPFS and handling transactions on behalf of each node.
3. Ethereum validates the transaction, confirming the registration of the JSON artefact into the chain.
4. Once the transaction is confirmed, the *marketplace.mjs* script uploads the JSON file to the IPFS via the IPFS API (port 5001).
5. After uploading to the IPFS, the JSON artefact is sent to the next corresponding actor/actors (in this case, the experimentation manager).

Table 1 – Containers and ports

Node	Container name	IP address	Ports	Function
Evol_Ctr	evol-ctr-py	172.16.239.11	6004	Builds controllers combining modules Sends controllers for experimentation
	evol-ctr-ipfs	172.16.239.12	4001 8090:8080 5001	Libp2p swarm connection HTTP gateway and read-only API IPFS API
	evol-ctr-mkp	172.16.239.13	8545 3000	JSON RPC connection to Ethereum Marketplace HTTP API
Exp_Mg	exp-mg-py	172.16.239.21	6001	Receives controllers Passes controllers and experiment list and parameters to the Digital Twin
	exp-mg-ipfs	172.16.239.22	4001 8091:8080 5001	Libp2p swarm connection HTTP gateway and read-only API IPFS API
	exp-mg-mkp	172.16.239.23	8545 3000	JSON RPC connection to Ethereum Marketplace HTTP API
DT	dt-py	172.16.239.31	6002	Runs experiments
	dt-ipfs	172.16.239.32	4001 8092:8080 5001	Libp2p swarm connection HTTP gateway and read-only API IPFS API
Cur_Ctr	cur-ctr-py	172.16.239.41	6003	Decides if a controller is protected
	cur-ctr-ipfs	172.16.239.42	4001 8093:8080 5001	Libp2p swarm connection HTTP gateway and read-only API IPFS API
	cur-ctr-mkp	172.16.239.43	8545 3000	JSON RPC connection to Ethereum Marketplace HTTP API

The marketplace SC is responsible for registering artefacts generated throughout the evolution and experimentation process into the blockchain, ensuring traceability, security, accessibility and scalability. While the private IPFS network stores artefacts and provides scalability while ensuring availability, the Ethereum blockchain keeps track of all transactions. It constitutes a historical record of file versions, upload times and other metadata.

To achieve this, file registration is managed by *Marketplace.sol*, an SC laid out to capture meaningful information to ease the identification of files and browsing the register optimally. To this end, a structure named *File* has been created in the *Marketplace.sol* SC, whose elements are listed below:

- File number *(uint256)*: number of the file uploaded to the marketplace chronologically.
- CID *(string):* the file's content ID, computed before being uploaded to the IPFS.
- Name *(string):* name of the file.
- Object type (*string*): type of the object uploaded: controller, experimentation report etc.
- Uploader *(address payable)*: Ethereum address of the node registering the file.

A mapping is then assigned to the file structure and the total number of files uploaded to the marketplace, which keeps track of files in the system. This number is, in turn, saved in the chain as the file number attribute.

It is worth noting that while the *id* of a controller, as described in Section 3.1, represents a specific instance of a controller, the CID is a unique cryptographic hash that represents a file and does not change as long as the contents of the JSON stay the same (i.e. when the evolution is run multiple times).

This PoC also includes the automatic deployment of a controller as a REST service. The code for this part can be found in the folder *ctr_deployment* [7]. For this purpose, a YAML file is written according to the

OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) [10], as it is recommended in the build-a-thon proposal [11]. The TOSCA file is then parsed and executed by using the orchestrator xOpera [12]. With the CID of a controller as input, the following task is performed by the orchestrator (Fig. 17):

- Fetch the JSON representation of the selected controller from the marketplace.
- Create an instance of the class controller based on the JSON representation.
- Deploy a REST service in a docker container.

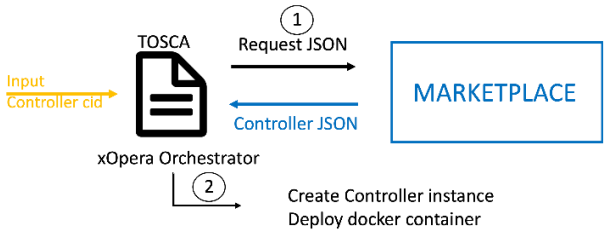


Fig. 17 – Representation of controller deployment

For the development of the REST service, we used the Python tools Flask and Gunicorn (Section 3.1.4). Once the service is deployed, it replies to HTTP GET requests (Fig. 18). The HTTP response is a JSON containing the controller's information and the output based on the request's input (Fig. 19).

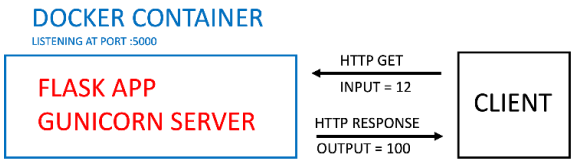


Fig. 18 – Controller REST service representation

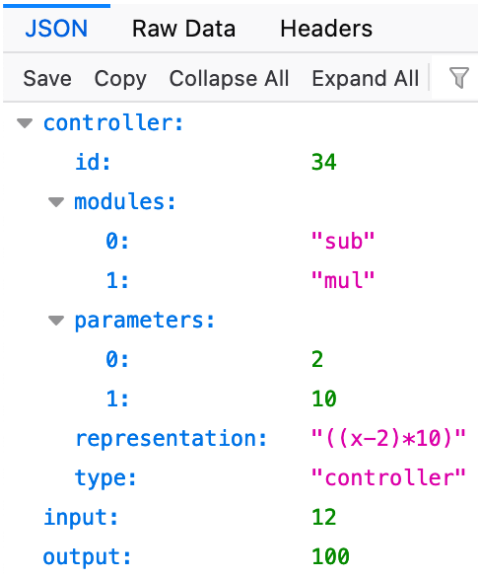


Fig. 19 – JSON response by the controller REST service

The controller implemented is relatively simple, and the REST service is not a closed loop (we already implemented a closed loop for this contribution in the evolution process). However, the present section proves the feasibility of deploying a controller with just its CID, getting its representation from the marketplace. The CID of the controller with id 34 is the default input in the file *inputs.yaml* inside the *ctr_deployment* folder [7], although any other controller's CID can replace it. As explained in Section 3.2, this CID will remain the same as long as the JSON representation of the controller is the same, regardless of how many times the evolution is run. This feature ensures a "check-point" to which the service can always return.

The configuration parameters for the presented PoC are as follows:

- Each controller is composed of two modules.
- The parameter of each module can only adopt two possible values: 2 and 10.
- Evolution is done by randomly combining pairs of modules, avoiding repetition of the same modules in the same order with the same parameters (that would yield an identical controller).
- The PoC stops after trying out all possible combinations.
- As for the curation controller criteria, the default parameters are used. As explained in Section 3.1.4, this means:
  - *Average*: protected controller if the result is greater than 500.
  - *Value*: protected controller if the result is smaller than 20.
- All docker containers run on a local machine and in a private network, as explained in Section 3.2.

This configuration aims to illustrate the proposed architecture's mechanics through a clear and quick demo easily reproducible in a local machine by anyone who clones the repository [7]. The closed-loop chronological order was explained in Section 2.1, and it is shown in Fig. 2. Fig. 13 shows the representation of the actual docker container nodes.

Fig. 20 shows the log messages of the evolution controller (its Python part), denoted as Ev_Py in Fig. 14. The output shows the JSON representation of each controller that the Ev_Py node sends to Ev_NJS. The JSON is forwarded to Ev_IPFS, and its response, the CID assigned to each controller, is sent back to Ev_Py by the Ev_NJS. With the printed CID, it is possible to access the controller JSON representation in the marketplace, as shown in Fig. 21 for the controller with id 34. It is interesting to change the port in the address to access a different node (Fig. 22). Port 8091 corresponds to the Exp_IPFS node and the 8092 to the DT_IPFS (see Fig. 15 and Table 1). This proves that the information about each controller is reproduced in all the IPFS nodes.

```
evol-ctr-py {'file': '{"type": "controller",
"id": 32, "modules": ("sum", "pow"],
"parameters": [10, 10], "representation":
"((×+10)^10)"}'}
evol-ctr-py {'cid':
'QmcRhK308tW7m6rQAxrargU6a8x7X9Smyxd7rDWV7x1
UR'}
evol-ctr-py {'file': "("type": "controller",
"id": 33, "modules": ["sub", 'mul"],
"parameters": [2, 2], "representation":
"((x-2)*2)"} '}
evol-ctr-py {'cid':
'QmY9KpCHhefRCCBQowpCqZoDacRBycyMQBLgris534F
U8V'}
evol-ctr-py {'file': '("type": "controller",
"id": 34, "modules": ["sub", 'mul"],
"parameters": [2, 10], "representation":
"((x-2)*10)"}'}
evol-ctr-py {'cid':
'QmP5ffwUvKwUDpoMCSvdw6rwfy3r2CoZAgDJJtiY8h8
4B40}
evol-ctr-py {'file': '("type": "controller",
"id": 35, "modules": "sub", "mul"],
"parameters": [10, 2], "representation":
"((x-10)*2)"}'}
evol-ctr-py {'cid':
'Qmww2qBK9TsRskivWXKryG6caFq7fx$99trB5RJRMKB
5Dx'}
evol-ctr-py {'file': '("type": "controller",
"id": 36, "modules": ["sub", "mul"],
"parameters": [10, 10], "representation":
"((x-10)*10)"}'}
```

Fig. 20 – Log messages of the Ev_Py docker container

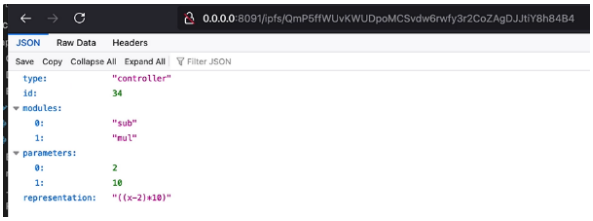


Fig. 21 – Access to the JSON representation of a controller in the marketplace via its CID. The marketplace node that is being consulted is Exp_IPFS.

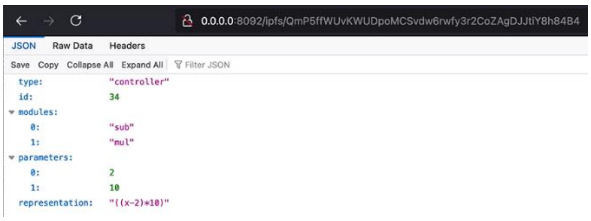


Fig. 22 – Access to a controller JSON representation in the marketplace via its CID. The marketplace node that is being consulted is DT_IPFS.

The next logical step is to check the Exp_Mg output (Fig. 23), including the information sent to conFig. DT_Py, as well as its response. Experiment results are also uploaded to the marketplace. Fig. 24 shows the Exp_NJS output containing each experiment's CID and information related to the Ethereum SC. The output of the *marketplace.mjs* script indicates that a new artefact was received, registered to the chain and uploaded successfully to the IPFS through the *Marketplace.sol* SC. This is indicated by printing the result of uploading the file to the IPFS (the path and CID generated for the file) and the HTTP POST request sent to ports 6004 and 6003 of the Evol_Ctr and Cur_Ctr, respectively. The JSON representation for each experiment has a CID that allows access to it in the marketplace (Fig. 25).

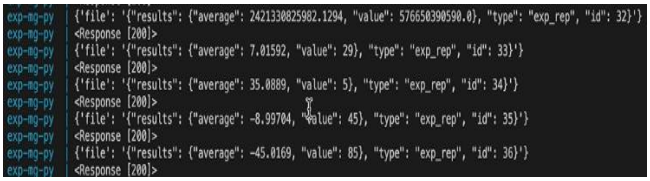


Fig. 23 – Log messages of the Exp_Mg docker container

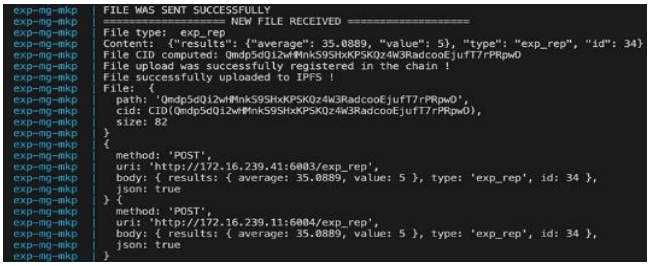


Fig. 24 – Log messages of the Exp_NJS docker container

Fig. 25 – Access to the JSON representation of an experiment in the Marketplace via its CID

To check if the controller has been promoted to the protected category requires looking at the Ito Cur_Py log messages (Fig. 26). Focusing once again on the controller with id 34, it is possible to access the decision of the curation controller in the marketplace using the corresponding CID, as shown in Fig. 27. This controller ends up with the category of protected for the *value* experiment, but not for *average*.

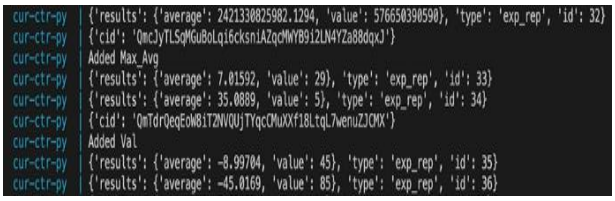


Fig. 26 – Log messages of the Cur_Py docker container

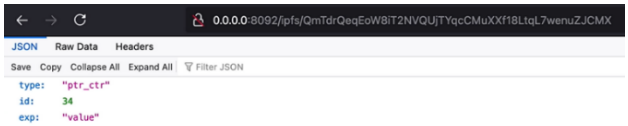


Fig. 27 – Access to the JSON that registers if a controller has the protected category in the marketplace via its CID

As explained in Section 3.3, after the evolution process has occurred, it is possible to deploy a REST service based on a controller that, when receiving HTTP GET requests with a float input, produces an HTTP response with a JSON that contains the information about the controller and the output for the sent value (Fig. 19). The final REST service listens at port 5000 and uses the example controller with id 34. Fig. 28 and Fig. 29 show the response of two GET requests with different float values as inputs. The JSON the service produces contains information about the deployed controller and the output value for each request.

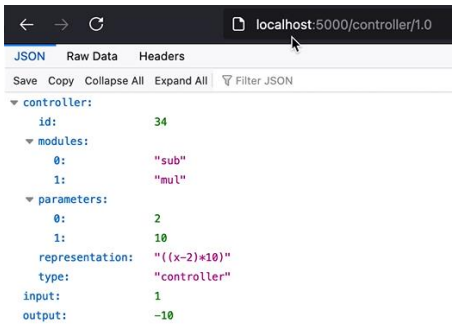


Fig. 28 – REST service response for the value 1.0

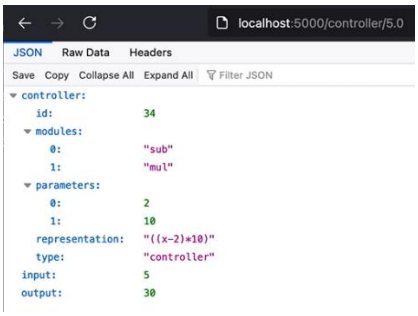


Fig. 29 – Example REST service response for the value 5.0. Deriving new use cases based on link prediction algorithm.

3.5 Discussion

This section focuses on laying the skeleton for an evolution framework based on the concepts described in [5], leaving details regarding each component for further development. To this end, simplified implementations of modules and controllers are used for the PoC and random combinations instead of a more complex evolutionary algorithm. However, specific improvements on each of these examples can be made relying on the presented work, with little or no change in the architecture. That key feature makes this PoC valuable for the future development of the use cases for ANs.

Another key focus of the proposed approach is the assurance of security and traceability of the evolution process through the implementation of a decentralized marketplace. The design and operation of the marketplace have been described in this paper. We believe it represents a strong feature of our approach as well, as thanks to the implementation of Ethereum and the IPFS it successfully provides security, auditability, privacy and scalability through decentralization and a distributed means of storage. Furthermore, the choice to use the popular and widely-adopted open-source technologies Ethereum and the IPFS facilitates its future development and accessibility.

On top of security and auditability, using JSON as the data format for exchanging the controller and experiment information in the marketplace enhances the explainability of the process since it is a human-readable format. Moreover, the deployment based on docker makes the PoC portable, and by using open-source tools and providing the source code, we have ensured its future use and reproduction.

4. USE CASE 3: DERIVING NEW USE CASES BASED ON LINK PREDICTION ALGORITHM

It is important to investigate the hidden use cases for ANs to effectively extract how to use AN concepts in future networks. An effective way to do so is by using recent use cases to predict the possible ones.

Instead of manually analysing the existing use cases, we can benefit from the huge advancements in computational intelligence, Artificial Intelligence (AI) or Machine Learning (ML). Nevertheless, for this purpose, we need a dataset that contains information about the recent use cases.

First, we analyse and automate the process of getting information (text and graphs) as described in [4]. The network structure and entities are captured through the data presentation, and a model is proposed to predict the links. The rest of the section discusses text and document parsing, figure parsing process, graph database representation and link prediction algorithms.

4.1 Use case design

We utilize the docx library [13], a programming library based on Python programming language, to process the use case document [4]. Information needed is extracted from the document using the library. For instance, “category, use case category, notes on use case category” represent the same information and, thus, are referred to as the” category of the use cases”.

Each use case's requirements are then parsed and presented as paragraphs after the table of each use case. Information extracted includes the use case number, requirement numbers and priority level, of which there are three:

Critical: important and required for the implementation of use cases;

Expected: possible requirement which would be important but not necessary to be fulfilled;

Added value: possible requirement which would be optional to be fulfilled (e.g. by an implementation), without implying any sense of importance regarding its fulfilment.

Fig. 30 represents the actors' interaction, while Fig. 31 represents possible components. To extract the information from figures, we convert it to a machine-readable format and then extract the desired information. To address the issue, several tools were employed. The first tool used is PlantUML [14] to create a sequence diagram.

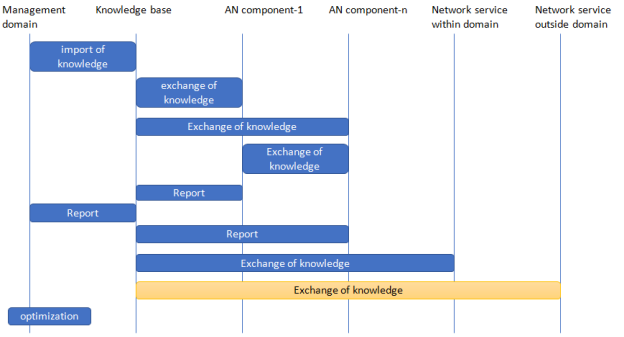


Fig. 30 – Actors interaction diagram

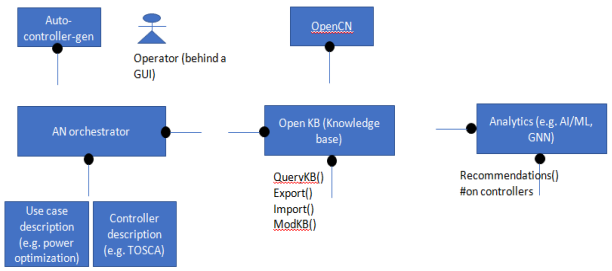


Fig. 31 – Possible component diagram

PlantUML is a tool to make several types of diagrams using the the UML format. By using this tool, we can create sequence diagrams, and the result on our first try was as shown in Fig. 32.

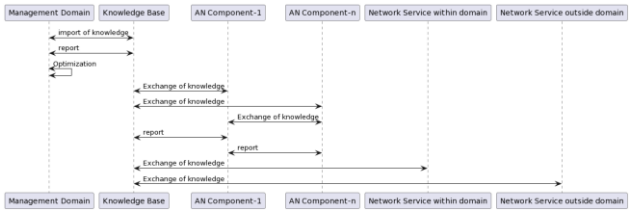


Fig. 32 – First attempt at using PlantUML

By comparing Fig. 32 and Fig. 33, we can see that this representation is most like the desired one. Still, the crucial disadvantage, the optimization function of the management domain here, is represented as a self-massage. Still, it is a function inside the management domain component in real time.

We then tried another representation this time, tried another type available in PlantUML, and the result was shown in Fig. 33.

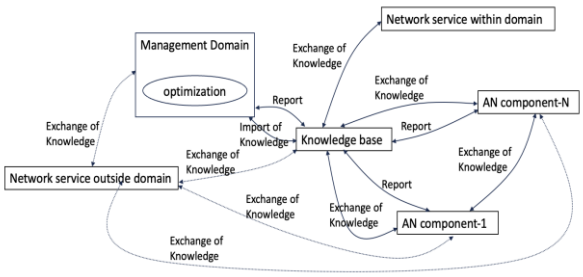


Fig. 33 – Second attempt at using PlantUML

This representation is sufficient if we do not want to see a visual representation; here, we overcame the problem of representing the function, but we have a bad visual, not like the one we want to be consistent.

In the final attempt, we used another tool to make the representation, Draw.io [18], a GUI-based tool from which we can extract the XML or any other machine-readable format; the result is as shown in Fig. 34.

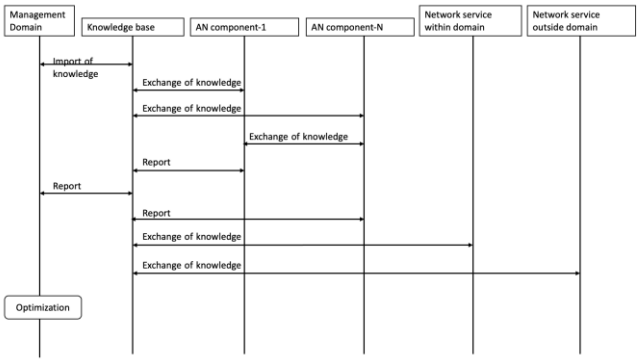


Fig. 34 – Final try using Draw.io

The following metadata is added:

Class: to represent the class of the item, whether it was a component, function or the type of the relation, this field is added to all items in the figure;

Value: the item's name or the relation to all items;

Parent: to contain the parent component ID that contains this function, only included in the function items.

After successfully representing the figure in XML, which is a machine-readable format, we can use Python and its library XML [13] to parse the figure; and we ended up with Table 2.

Table 2 – Relations table

	source	target	message	class	id
0	Knowledge base	AN component-N	exchange of knowledge	critical relationship	UC001
1	AN Component-1	AN component-N	exchange of knowledge	critical relationship	UC001
2	Knowledge base	AN Component-1	Report	critical relationship	UC001
3	Knowledge base	AN Component-1	exchange of knowledge	critical relationship	UC001
4	Management domain	Knowledge base	import of knowledge	critical relationship	UC001
...	...	...	...	...	...
125	AN_1	AN_3	negative acknowledgement for the unicast reply	critical relationship	UC040
126	AN_1	AN_2	Configuration and context setup request requir...	critical relationship	UC040
127	AN_2	AN_1	context setup response	critical relationship	UC040
128	AN_1	AN_2	AN_1 configures the network underlays for rout...	critical relationship	UC040
129	AN_1	AN_2	AN_1 pays AN_2 for the service	critical relationship	UC040

Also, we have a dictionary that contains the functions and their corresponding parent actors. Now we have completed parsing the document, and Fig. 35 represents the major steps taken.

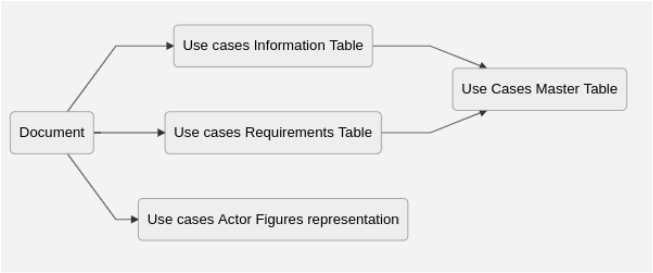


Fig. 35 – Major steps of the data parsing process

After this, we need a way to convert this into a type of dataset to run machine-learning algorithms on it.

4.2 Graph database representation

There are several graph databases; we chose Neo4j [15], one of the most popular. It is more convenient to represent the network with a graph database. In the graph database, we represent the data in the form of nodes and relations (the most relevant format for network structure).

We made some iterations to represent; the first try was based only on the relations table, which contains the components in the relations table with 130 relations only, but this representation has some drawbacks unique to those use cases that have figured to represent it on the document.

Next, each use case is read carefully and the true relations and functions inside the graphs extracted, including the hidden relations in the requirements and descriptions of each use case. The final product is the reference code, which is sufficient to represent the use cases, now we have +500 relations and +300 components. Fig 36 shows the graph representation of Neo4j. Now we have represented the data, the next step is to run the link prediction algorithm on it to be able to investigate new use cases.

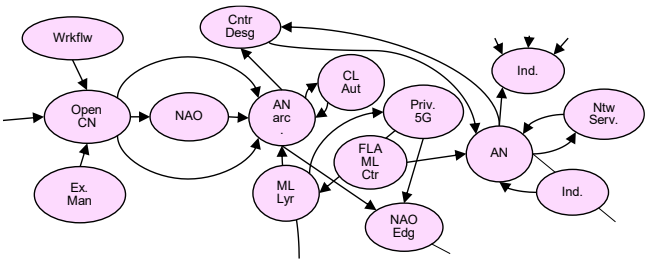


Fig. 36 – Graph representation

4.3 Use case PoC implementation

We can consider the link prediction pipeline as a classification problem to determine whether a certain relation exists between two nodes. The classification accuracy is improved if we calculate something that measures how likely these two nodes are to have a relation in between. The current representation needs some numeric properties.

And the pipeline of the link predictions contains several steps. The first step is to project the graph database in the memory; this is done by using the native projections in neo4j. The next step is the addition of node properties.

Node embeddings. We need specific embedding for each node, like what happens when we decode a categorical feature in ordinary data preprocessing. Node embedding algorithms compute low-dimensional vector representations of nodes in a graph. These vectors, also called embeddings, can be used for machine learning (see Fig. 37).

We used the fastRP algorithm to generate low-dimensionality node embeddings (vectors) through random projections from the graph's adjacency matrix. The FastRP algorithm was selected because it is mature to be used in production quality and works for directed and undirected relationships.

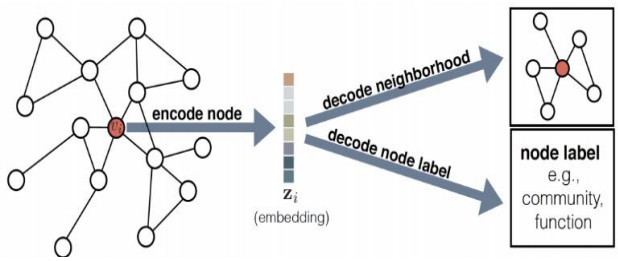


Fig. 37 – Graph representation

Centrality. Centrality algorithms are employed to determine the importance of distinct nodes in a network.

We used the degree centrality algorithm, which works well in directed and undirected relationships. The degree of centrality measures the number of incoming/outgoing relationships from a node (Fig. 38).

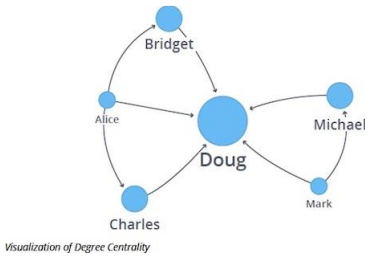


Fig. 38 – Example of centrality

Community detection: Community detection algorithms evaluate how groups of nodes are clustered or partitioned and their tendency to strengthen or break apart. We have used the Label Propagation Algorithm (LPA), as it works well for directed and undirected relationships.

The LPA is a fast algorithm for finding communities in a graph. It detects these communities using network structure alone as its guide, requiring neither a predefined objective function nor prior information about the communities. Now we have added all the properties that may help the algorithm more and more, the next step is to calculate combined features using these properties. The properties created on nodes in each potential link are combine using L2, Hadamard and cosine function.

Training and prediction: When all the pipelines were ready, we trained algorithms in the dataset, and after tuning hyperparameters, we found that the RF model behaved best in all use cases. Then, we needed to test the models in all scenarios:

First scenario: We used only the 130-relation dataset, which resulted from using only use cases with figures.

Second scenario: We used the full representation of the reference code with +500 relations and +300 nodes.

Third scenario: We used the second scenario but added the actors' functions as properties in these nodes.

Table 3 shows the experimental results. From the results, the third scenario is the best scenario, and when we made the prediction we found 6954 new links with more than 99% of probability to be a true relation.

Table 3 – Experimental results

Representation	Best model	Test accuracy (AUCPR)
Scenario 1	RF	88%
Scenario 2	RF	73.6%
Scenario 3	RF	86%

Draw.io, with many customizations, is the best tool for making make more representative figures. The best algorithm for link prediction is the Random Forest (RF) model, with 86% AUCPR accuracy in the full reference code. The Area Under the Precision-Recall Curve (AUCPR) is a single-number summary of the information in the Precision-Recall (PR) curve.

Future studies need to increase the coverage of representation of the use cases in the reference code e.g. to convert all functions into properties of nodes.

5. USE CASE 4: A LOW LATENCY CLOSED LOOP FOR ROBOTIC GRASPING

This section analyses the use case [16], "Slip Detection (and Force Estimation) and Object Detection" in robotic grasping. Primarily, universal grasp action requires object picking with compensatory grip force control to avoid gross object slippage. The secondary task for hand manipulation is identifying the object type across different properties, primarily detecting the class. Using the MEC testbed, a low latency closed-loop system is developed between the haptic hands/algorithms and a real robotic hand (Allegro Hand) [16]. Fig. 39 presents the low latency closed loop for the robotic grasping problem. The robotic hands perceive the environment using various sensors type, such as intrinsic and extrinsic sensors, providing the environment's posture, object contact and physical information [17]. The haptic hands consist of a haptic-feedback system which can be classified into the kinesthesis and cutaneous systems [18]. Kinesthesis deals with an individual's body movement perception, enabling effective movement and motion. The cutaneous deal with touch perception, detecting pressure and vibration that enables gripping and avoids slippage in the process. The details of the slip detection, force estimation and object detection considered in this paper are provided in the following subsections.

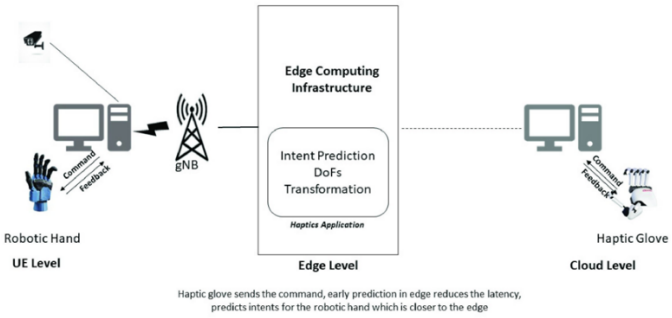


Fig. 39 – The low latency closed loop for the robotic grasping problem

5.1 Use case design

Slip detection is a crucial component of robotic grasping. A robotic arm can sense when its grasp is slipping. This is important for robotic arms to adjust their grip and maintain a secure hold on the object. Slip detection is achieved using tactile sensors or through the gripper's joints' force feedback, which detects changes in pressure and friction as the robotic hand interacts with the object. This data is then used to identify when the object is slipping from the grasp, and the robot can adjust its grip accordingly. When the object begins to slip, the robot applies more grip force, and when it is secure, less force is applied. This helps the robot maintain its grasp of the object, allowing it to manipulate it more precisely. Slip detection in robotic grasping ensures that the robotic arm can accurately grasp and keep objects in place. For this, a low-latency machine-learning model control approach can be used. The machine-learning model can be trained on a dataset containing objects, their physical properties and the desired robotic grip parameters. The model can then detect when the object is slipping out of the robotic grip, allowing it to adjust its grip parameters accordingly. This approach ensures that the robot can quickly and accurately detect slipping objects and make the necessary changes. Additionally, the low latency of the machine-learning model control approach allows the robot to respond to slipping objects promptly, ensuring that objects are not dropped or damaged during the robotic grasping process.

Object recognition in robotic grasping is a complex field of study addressed by many research projects in robotics and computer vision [17-22]. At its core, object recognition identifies a specific object within an image or video frame. This task is typically accomplished by applying specialized algorithms such as convolutional neural networks or other machine-learning techniques. In recent years, object recognition has become increasingly

important for applications in robotics and automation. Haptic force-based object recognition is a method of robotic grasping that relies on tactile feedback from physical contact. By measuring the forces applied to the object, the robot can identify the shape and size of objects, and it can determine the best way to grasp and manipulate it. This technique can be used in a variety of circumstances, such as when the object is occluded from view or when it is too small or too large to be detected by vision-based object detection. Compared to vision-based object detection, haptic force-based object detection offers greater accuracy and reliability, as it is affected by neither lighting conditions nor the presence of occluders in the environment. Additionally, it is not limited to detecting flat or rigid surfaces. It can even detect the texture and properties of an object, such as its weight or material composition.

Fig. 40 presents the MEC testbed for the model inference investigation. The two cases considered are presented as in the following two subsections.

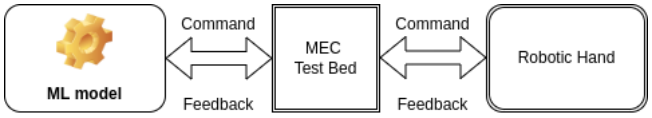


Fig. 40 – Model inference using MEC testbed

5.1.1 Case 1: Slip detection

Any combination of the following individual features forms the basis of the feature set for the model to train upon. All individual features and possible combinations were fed into the learning model during experimentation.

Feature engineering – Model features space:

1. Joint Force F_t
2. Joint Position,Θ_t
3. Force Derivative, ΔF_t
4. Position Derivative $\Delta \Theta_t$

The target label for the slip event and crumple event are transformed into four possible combined classes for the model to predict the slip and crumple state at any given time step as a combined output, as depicted in Table 4.

Table 4 – Transformed events label for multiclass prediction

Events		Resulting events
Slip	**Crumple**	
No	No	0
Yes	No	1
No	Yes	2
Yes	Yes	3

Methodology: For the defined problem statement of detecting the slip event, as well as the crumple event during the grasp-lift phase of object picking, the time-series readouts data from the 16 joints of the gripper is transformed into the required input format of the LSTM-based model presented in Fig. 41. The data is transformed by applying a sliding window to the time-series dataset with window size equal to the number of previous observations before the model predicts the output for the next time_step. The transformed dataset is shuffled to avoid biased learning.

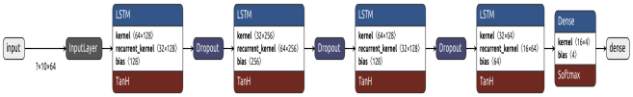


Fig. 41 – LSTM model for slip detection

- Input shape: (n_samples, n_previous_time_steps, n_features) The build model was tried on different features combinations (n_features: [16 – F_t, 16 – Θ_t, 16 – ΔF_t, 16 – $\Delta \Theta_t$])
- Output_shape: (n_samples,n_output) The model predicts a class out of n_outputs (one-hot encoded) classes (n_outputs: [0,1,2,3]) corresponding to each transformed target class.

5.1.2 Case 2: Object recognition

For this case, the following individual feature forms are used for training the model.

Feature Engineering – Model features space:

1. Joint Force, F_t
2. Joint Position, Θ_t
3. Mass, M_{kg}
4. Joint Force x Joint Position, F_t x Θ_t

The target labels can be segregated into 13 unique object types (class: unique shapes and sizes, properties) and 6 unique object categories (same shape objects).

- Classes: 13 objects

- Class categories: 6 object categories.

The dataset is preprocessed further based on the unique categories to get the extracted dataset which is then fed to the classifier. Any combination of the above individual features forms the basis of the feature set for the model to train upon. All individual features and possible combinations were fed to the learning model during experimentation.

Methodology: The raw dataset was highly imbalanced for the defined problem statement of class recognition. Regarding class imbalance, the raw dataset with unique object classes was processed to get the balanced dataset to avoid the model's biased learning due to imbalanced samples of data of individual classes. The training dataset is generated by extracting each class dataset equivalent to the category with the least sample in the raw dataset.

- Input data: (n_samples,n_features) The build model was tried on different features combinations (n_features: [16 – F_t, 16 – Θ_t, 1 – M_{kg}, 16 – F_t x Θ_t])
- Output_shape: (n_samples,n_output) The model predicts a class out of n_outputs (one-hot encoded) classes (n_outputs: 13 classes) corresponding to each transformed target class.

5.2 Use case PoC implementation

For case 1, we considered 10 previous time steps as the length of the observed window for the model to predict the event class. The model is trained with an Adam optimizer with a default learning rate of 0.01 and categorical_crossentropy as a loss function.

Furthermore, an RF is used for case 2. As the name implies, a random forest consists of many individual decision trees that operate as an ensemble. Each tree in the random forest spits a class prediction, and the class with the most votes becomes our model's prediction.

Illustrative results for both use cases across the combination of raw and constructed feature trials for slip detection and object recognition are summarized in Table 5. It can be observed that there were minor differences in the classifier behaviour across the feature trials for the object recognition case, but the overall performance was highly accurate due to well-constructed features and the cleaning of noise from the raw data performed during the feature engineering stage. The model trained for slip detection cases also performed reasonably well. Nevertheless, a significant increase in accuracy can be observed for the feature set involving derivative features of raw data. Thus, the derivative features, like $\Delta\Theta, \Delta F$, reasonably impact the model performance for inference of resultant slip and crumple events during the grasp-pick phase of lifting the objects.

Table 5 – Result table for both use cases

Method	Features	Slip detection		
		Train	Validation	Test
LSTM	Θ, F	79.6	80.4	80.6
	$\Delta F, \Delta\Theta, \Delta F$	86.4	84.2	85.6
RF		Object detection		
		Train	Validation	Test
	Θ, F	96.1	97.5	96.9
	$\Delta F, \Delta\Theta, \Delta F$	98.1	99.5	98.9

6. USE CASE 5: MONITORING OF THE MOST SIGNIFICANT KPIS AND THE AUTOMATIC DETECTION OF ANOMALIES

In recent years, more users have been connected to mobile networks [23, 24]. With the onset of 5G, the number of IoT devices that will use the 5G RAN will be huge [25, 26]. This causes more and more 5G RAN nodes to be deployed, thus causing network density. The challenge is now to have an improved network capacity with a fully-functional network without any outages. Since the number of 5G RAN nodes that collect data on the network performance will greatly increase, the volume of metrics to be managed and analysed will rise. In such scenarios, each node collects hundreds of the most important network performance indicators, known as Key Performance Indicators (KPIs).

Data analysis of this KPI data becomes more critical than ever since new functionalities are implemented in mobile networks, allowing real-time modification of available network resources and services. This demands an automatic system that detects behavioural changes in KPIs before operators complain about the degraded quality of experience.

In this way, network operators and domain experts could make decisions to correct this anomalous behavioural effect if necessary. In addition to anomalies, network behaviour patterns may change over time, exhibiting seasonality. Therefore, detecting anomalies over time impacts the quality of the experience operators perceive. Thus, the

earliest anomaly detection is critical in 5G RAN for the above-mentioned reasons.

Anomaly detection is actively and heavily researched. There is abundant literature on the subject and many machine-learning techniques that could be used to detect anomalies, such as regression using ARMA, ARIMA, SARIMA or clustering techniques like K-Means, DBSCAN or classification techniques like KNN, random forest etc.

In this paper, we focus on the use of unsupervised deep-learning methodology, LSTM-VAE, on periodic time-series KPI data, which are used as design elements for building the FG-AN-use-case-41 [4]

The KPI data is for the 5G RAN in the (timestamp, cell_id, kpi_id, value) format. The 3GPP standard [27] categorizes the KPI areas into the following:

a. Accessibility KPI – RRC setup SR, ERAB setup SR, call setup SR etc.
b. Retainability KPI – call drop rate, call setup complete rate etc.
c. Integrity KPI – radio network unavailability rate etc.
d. Availability KPI – HHO SR, HO SR, Inter-RAT HHO SR etc.
e. Mobility KPI – service UL/DL throughput etc.

The KPI data can be collected and saved through the network management system at a time interval of every 15mins/1 hours without impacting the system performance.

The key KPI areas in 5G RAN that can be monitored for service impacts includes:

a. Call access – low access rate, increased call drop rate
b. Throughput – DL throughput, UL throughput
c. Cell availability – In this paper, we will describe the unsupervised deep-learning methodology to forecast 5G RAN KPI anomalies much in advance.

6.1 Use case PoC implementation

An anomaly is defined as an entity that deviates from normal, standard and expected behaviour. Traditional methods for performance anomaly detection are based on static thresholds and rules which work well only for simple Key Performance Indicator (KPI) monitoring. Unfortunately, finding the appropriate threshold levels for complex KPI monitoring is difficult when significant volatility stems from different usage patterns. Hence, it is mandatory to study the previous KPI behaviour to decide whether the current one is anomalous or not.

A KPI data can consist of both seasonal and non-seasonal components. Seasonal patterns may change or shift over time. Thus, the KPI patterns must be characterized and updated over time. This demands the anomaly detection system to adapt automatically to new seasonal patterns.

An anomaly detection system enforces the following requirements to be fulfilled by it:

a. Quicker detection of anomaly: Detect all the KPI anomalies as soon as possible to ensure that operators can take corrective action sooner, thereby reducing the users' affected time.
b. Automatically adapt to new behaviours: If the KPI's normal behaviour changes over time, the system must automatically adapt to the new behaviour.
c. Minimize anomaly errors: The system reports only those abnormal behaviours of the KPI data that do not fit any previously identified patterns. This minimizes the wastage of resources on non-existent troubles by the network operator.

An anomaly forecasting system can reuse the same unsupervised deep-learning methodology, LSTM-VAE, for forecasting the 5G RAN KPI data in future time points. This data can be fed as an input to the anomaly detection system to auto-classify the KPI anomalies.

The forecasted KPI anomalies can help in the identification of 5G RAN KPI degraded managed objects (Cell/BTS). This further assists the network operator in performing preventive maintenance to do the necessary troubleshooting much in advance and increase the network capacity (increase Cell/BTS in each area etc.), ensuring the quality of assurance is kept intact.

In this paper, we will describe the steps which can be adopted to develop an automatic anomaly detection and forecasting system using the unsupervised deep-learning technique LSTM-VAE, and we will also describe the design of such a system in the form of graph representation, which was produced from the use case FG-AN-usecase-41 [4].

Fig. 42 architectural flow diagram describes major elements of the mentioned anomaly detection and forecasting system.

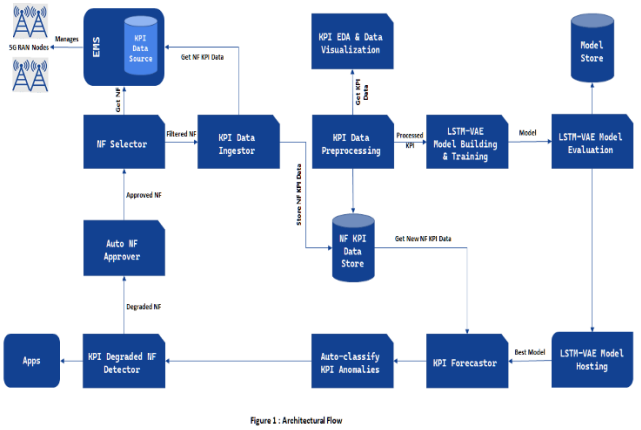


Fig. 42 – Architectural flowchart

The key building blocks of this system are as follows:

a. KPI data store: This refers to the data storage repository of the KPI data which is collected from the 5G RAN managed objects through the network management system managing the managed objects. This data store consists of the raw historical KPI data being collected and stored. This historical KPI data could consist of minimum 2-week old KPI data. As and when a new set of KPI data is available, it gets stored into this data store.

b. KPI data ingestor: This block ingests the KPI data from the data source to be used by the data preprocessing block.

c. KPI EDA and data visualization: Use statistical techniques to analyse and visualize the KPI data. Seasonal and Trend decomposition using Loess (STL) time-series decomposition techniques [38] can be used to separately view the trend, seasonality, residual of the time-series data.

d. KPI data preprocessing: The KPI data is usually considered as time-series data. Multiple KPI data pertaining to a given area can be grouped together to derive a feature forming multivariate time-series data, which captures the relationships among the constituent KPI data [28, 29].

The following preprocessing techniques are to be performed when dealing with multivariate time-series data:

- Volatility: Check for volatility/fluctuation in the time-series data. This can be done using the simple statistical techniques like: mean, variance and standard deviation.
- Normalization: Not all KPI data is following a single unit of measurement. Hence we need to normalize the time-series data onto a scale of [0-1] ensuring that the shape of the data is preserved. For example, Z-Score Normalization which has Mean=0 and Std Dev=1, Min-Max Scalar etc.
- Remove Volatility: Remove the increasing/decreasing volatility. Many machine-learning models (ex: ARIMA, VAR etc.) expect volatility to be constant. For example, divide each year data with standard deviation of that year data.
- Stationarity: Check if the time-series data is stationary. Any machine-learning prediction and forecasting works well only on stationary time-series data. For example, use ADF/unit root test (do this at the end of every preprocessing step).
- Transform non-stationarity to stationarity: If the given time series is not stationary, then transform the time-series data to stationary by removing the trend. For example, take the first difference to make the time-series data stationary.
- Remove seasonality: Next remove the seasonality/seasonal pattern from this resultant data. For example, Take the average of each month across the years and subtract each of the data points from their respective month's average.
- Train-test data split: Split the resultant data into a 70-20 training data and testing data split. The train data will be consumed by the model building and training block and the test data will be consumed by the model prediction block.
- Time steps for forecasting: Use of statistical techniques like Auto-Correlation Function (ACF) and Partial Auto-Correlation Function (PACF) to determine the time steps/time window size which can be used for model training and prediction. The ACF plot in Fig. 43 shows that a time step of 25 time points is more significant compared to the time step of 51 time points.

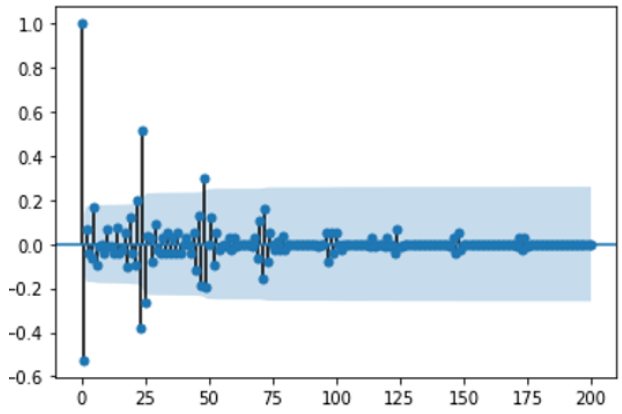


Fig. 43 – ACF of 5G RAN KPI data

e. Model building and training: Here we propose the use of the Long Short-Term Memory-Variational Auto Encoder (LSTM-VAE) deep-learning algorithm on the training dataset. The training dataset consists of the preprocessed KPI data and upon consuming it, the LSTM-VAE trains the deep-learning network with relevant activation functions and the parameters which are configured. Based on the configured training loss and validation loss parameters, the model trains the given system until the early stopping criterion is met. LSTM-VAE is effective on the multivariate time-series data when the time-series data exhibits nonlinearity.

f. Model evaluation: The trained LSTM-VAE model consumes the test data for the model validation and forecasting. Model validation and evaluation is inbuilt by calculating the MAE and RE. A lower MAE and RE suggests that the model for the given KPI time-series data is well and the model can be used further for forecasting and anomaly detection [30-32].

g. KPI forecast: This block uses the trained LSTM-VAE-based model to forecast the KPI data. This forecasted data can be fed to a dynamic threshold-based auto-classifier to classify whether the KPI data exhibits an anomaly [33-35].

h. Auto-classify KPI anomalies: This block uses the threshold derived automatically and classifies those KPI data points as anomalies exceeding the threshold [35, 36].

i. KPI degraded managed objects: This block consumes the anomalous KPI data points and counts the total number of KPIs which are anomalous for every managed object and then selects the top five managed objects which have high a KPI anomalies count and finally stores them onto the data repository.

As and when the new data is stored and based on the periodic schedule configured, the LSTM-VAE model is retrained with the newer historical dataset and constantly updates KPI anomaly thresholds, and it adds the newly forecasted KPI data onto the respective data stores. Fig. 44 describes screenshots of the part of the graph produced from the FG-AN-usecase-41 design [4] described as part of this paper.

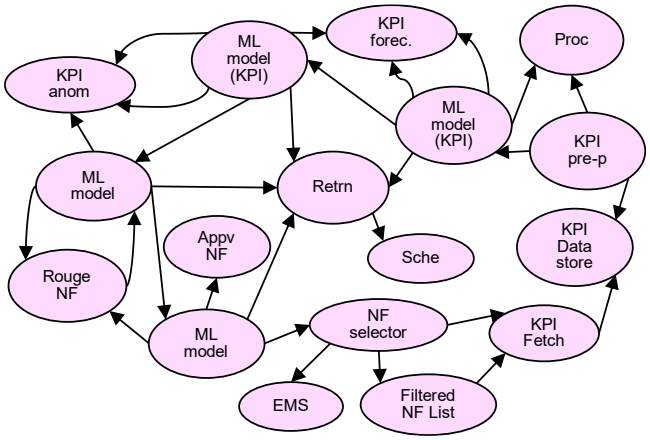


Fig. 44 – Graph design flow

In this section, we described a system for detecting and forecasting anomalies for 5G RAN using the LSTM-VAE deep learning-based methodology. The system consists of several stages that allow us to identify and distinguish service-impacting anomalies from occasional outliers.

We also suggested anomaly detection and forecasting of a group of multiple KPIs relevant to a given domain area from a particular node or a combination of KPIs from various nodes instead of single KPI anomaly detection.

The described system aids in timely fault identification and troubleshooting in networks to prevent network breakdowns.

7. USE CASE 6: DESIGN OF AUTONOMOUS AGENTS (WITH VARIED COMPETENCE)

This section presents the PoC on autonomous agents in the network case study. The case study has a level of autonomy for an autonomous system with estimation and judgement of competence as key criteria.

The autonomous system requirements are presented [4]. Among the requirement, the autonomous agent should determine and adjust the level of autonomy depending on the environment and type of task in contrast to the capabilities of the system in combination with policies to support. The capability will enable the system to dynamically adjust to the situation corresponding to human intervention requirements at run time or design. The main highlight of an autonomous workflow includes task speciation, understanding, feasibility, planning and execution. Other important considerations include the autonomous system's request for human support and subsequent learning.

A simple workflow scheme for autonomy with varying autonomy levels was discussed. This included task specification by humans and task understanding, a feasibility check, task planning and task execution by the system. Request for support from humans and control by humans can be to any of these workflow steps. Monitoring of performance levels by humans and learning by the system are added steps.

7.1 Use case design

For the use case design, we provide the graph-based design, where we consider sensors an integral part of all equipment, gadgets, mobility solutions, communication devices, medical devices, defence applications, athletics, electronics, computers, etc. The design is represented using a graph database.

The data detected from the network, in particular includes text data (logs), packet measurements, deep packet inspection, latency time measurements, packet loss measurements etc. The sensors are thus autonomous agents because they are created (instantiation) and managed (data collection, analysis and actions) with minimal human intervention.

Also, as part of the requirements for the use case, it is essential that ANs enable fault isolation through collaborative analysis of multiple sensors to identify the cause of the fault. Another aspect is enabling bug fixes without human interaction, such as discovering the correct version of software or hardware to update.

The creation of new agents with new capabilities, requirement collection and analysis of existing problems (and in some cases, recurring) in the network (RAN or base station failed) warrant the autonomous management of closed loops.

7.2 Use case PoC implementation

Based on the block diagram, a graph-based representation is created. In this PoC, we consider the validation of the actors and the relationships called out in the design using the graph database Neo4j and Python notebooks.

Autonomous controllers and data are considered as the main actors. The controllers consist of data stream sensors or data-delivering technology, while cloud-based (environment) methods are employed to distribute communication loads through relay nodes. Table 6 presents some of the sample actors and their relationships. Fig. 45 presents the graph representation created by the PoC using Python notebooks.

Table 6 – Actors' relationship

Actor-1	Actor-2	Description
Controllers	Autonomous	Sensors for data stream collection (or delivery) technologies.
Data	Environment data	Cloud-based (Environment) methods to distribute communication loads by relay nodes.

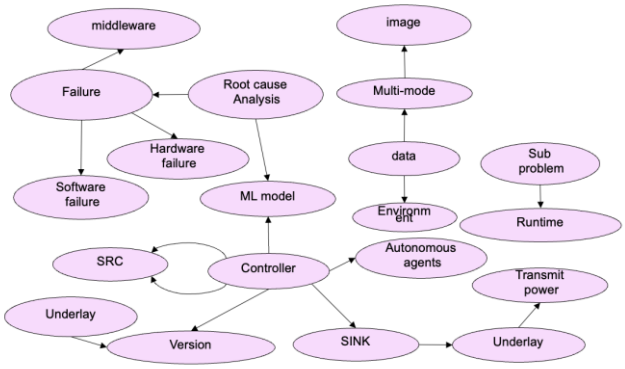


Fig. 45 – Graph database representation created by the PoC

This PoC solution addresses the following aspects while applying autonomous agents in systems:

- The failures detected in the autonomous systems are analysed and controllers (or agents) with corresponding capabilities or ML models are "matched" or selected, which can solve the root cause isolated during the analysis. This selection of agents is based on metadata collected as part of the design.

- Problems in the system are split into domain-wise issues, e.g. radio access network or core network issues and root cause analysis is done using existing models as described in Section 9.
- The capabilities of the controllers may include software or hardware parameter configurations, and upgrades, integrated into the underlay system as SRC (source) or SINK (application point) [37] for such configurations.
- Multimodal data analysis from the environment is suggested to study the performance of autonomous agents. This analysis may be used for optimization of existing agents or creation of new autonomous agents.

8. USE CASE 7: NETWORK RESOURCE ALLOCATION FOR EMERGENCY MANAGEMENT BASED ON CLOSED LOOP

This section examines network resource allocation for emergency management [4] and autonomous log troubleshooting based on closed-loop analysis.

8.1 Use case design

In the proposed system, all incoming KPIs in the network are scanned and slices are continuously configured for any new deployment. An RL agent optimises the slice configuration accordingly, using calculated rewards based on SLAs. Fig. 46 shows the proposed network emergency management system closed loop and graph representation.

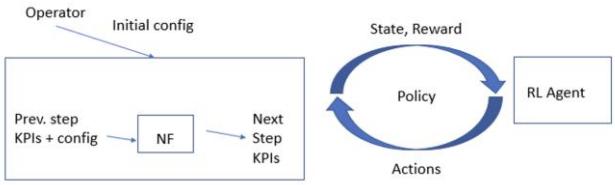


Fig. 46 – Design for network emergency management using a closed loop

This design reduces the problem to a single-cell resource optimization problem. A simulator is used to mimic the states of an actual network. Network operators may deploy initial network slice configuration values, which may be further optimized by the model later. The initial configuration values are considered as the "initial state". The throughput KPIs observed are mapped to the rewards of the RL agent. Actions for the RL agent are the change in the slice configurations. The system will take feedback from users and retrain the log anomaly detection to do self-correction.

9. USE CASE 8: AUTONOMOUS LOG TROUBLESHOOTING (WITH VARIED COMPETENCE) IN NETWORKS

This section presents the PoC on autonomous log troubleshooting (with varied competence) in networks. The case study focuses on fault prediction and isolation based on log analysis. The logs are generally unstructured with no standard format, making correlating logs across various vendors tedious.

Through historical logs, machine learning can learn and predict failure. The proposed solution entails scanning all incoming log traces continuously, automatically detecting and reporting system issues such as incidents with probable root cause.

9.1 Use case design

The system takes feedback from users and retrains the log anomaly detection to improve performance through self-correction. Fig. 47 and Fig. 48 present the system block diagram and graph representation, respectively. The main steps for the system include:

1. Logs collection from various open interfaces and NFs;
2. correlation and analysis of the collected logs;
3. performance optimization of log analysis mechanism.

9.2 Use case PoC implementation

Based on the block diagram, a graph-based representation is created. The actors and relationships are called out in this representation. This is then represented using a graph database. Anomalies and key anomalies form the important actors in the graph. A knowledge base created from log analysis and known issues curated by experts helps in root cause analysis for hitherto unknown situations in the log. Feedback from humans is an important aspect which helps minimise network errors due to the application of this analysis.

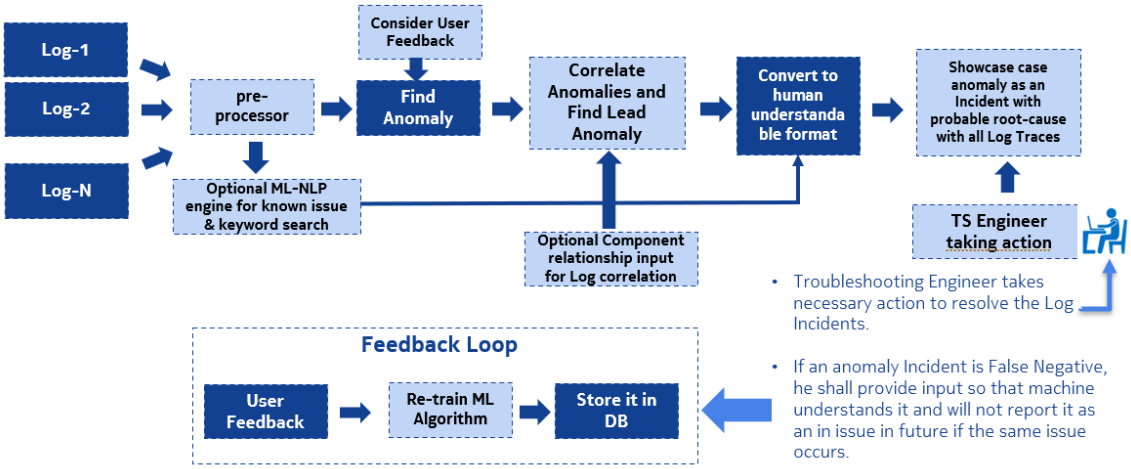


Fig. 47 – ML-based automated log troubleshooting

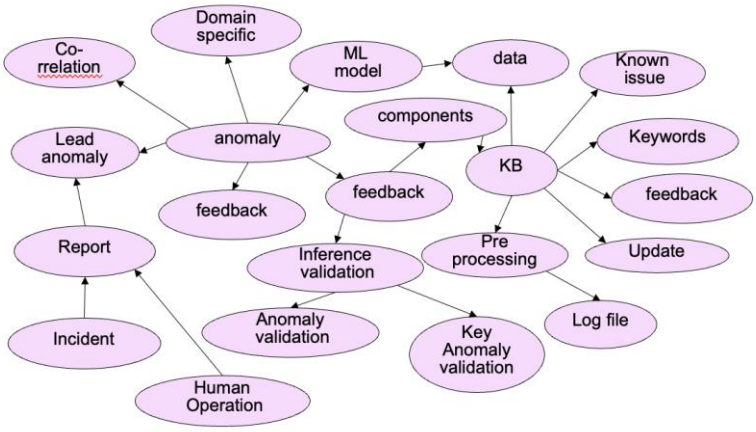


Fig. 48 – Graph representation of autonomous log troubleshooting

10. CONCLUSION AND FUTURE RESEARCH

Specific use cases were studied based on the use cases and concepts described in [4, 5]. A uniform representation using a graph format was applied to all use cases. The study included several use cases such as a knowledge import and export system, a secure and traceable evolution system for ANs, link prediction for inferring new relations between AN use case actors, "Slip Detection (and Force Estimation) and Object Detection", monitoring of the most significant KPIs and the automatic detection of anomalies, design of autonomous agents (with varied competence), and network resource allocation for emergency management based on a closed loop.

Each use case is described with a design and a PoC implementation using a graph database. Simplified implementations of modules and controllers are used for the PoC and random combinations instead of more complex evolutionary algorithms.

Derivation of graph representation of use cases from documents was described, and link prediction algorithms were applied. The best algorithm for link prediction is the random forest model with 86% AUCPR accuracy in the full reference code, and also we need to find a more enhanced way to represent the use cases in the reference code, i.e. to convert all functions to be properties.

Furthermore, we have described a system for detecting and forecasting anomalies for 5G RAN using deep learning-based methodology. The system consists of several stages that allow us to distinguish and track service-impacting anomalies from occasional outliers.

Some future research directions for use cases are presented as follows.

For future studies on evolution and blockchain for AN lines of work, the following ideas can be considered:

- Implement more complex definitions of modules and controllers for solving a specific problem in ANs.
- Add a list of experiments that is suitable for each use case, and use of standard design mechanisms such as modelling languages, e.g. Unified Modeling Language, for representation of the designs could be a future step.
- Implement more complex evolutionary algorithms than random combinations that can use the feedback that Evol_Ctr receives about the experiment results.
- Add the possibility of updating new modules after the evolution process has started.
- Implement more complex smart contracts that would allow for more complex access control, updating and deleting artefacts, as well as easier and more intuitive access to files.
- Add a more efficient detection and communication of updates within the system.
- Create a unified repository or marketplace for autonomous networks which can store the controllers.
- Data integration with autonomous networks, including data handling standards such as [37] are important.

Deriving new use cases based on a link prediction algorithm requires further enhancement to the method of representing the use cases in the reference code (i.e. to convert all functions into properties).

Additionally, future studies on deep learning for anomaly detection and forecasting in 5G RAN can focus on automatically using the KPI anomaly information to correlate anomalies of different KPIs to identify the root causes.

The logical next step for a low latency closed loop such as robotic grasping is to extend the proposed method to other rich tactile-based multi-fingered in-grasp manipulation tasks. Moreover, achieving several tasks (e.g. slip avoidance and object recognition) with one network for avoiding retraining on each task can be the next challenge.

Lastly, considerations to improve the coordination between the ITU and other entities to facilitate unified, interoperable customer services are of paramount importance.

ACKNOWLEDGEMENT

The International Telecommunication Union Focus Group on Autonomous Networks (ITU FG-AN) organized a “build-a-thon challenge” in 2022 to demonstrate and validate important use cases for autonomous networks, creating PoC implementations and tools in the process. The majority of the work in this study were done under the build-a-thon challenge.

REFERENCES

[1] P. H. Gomes, M. Buhrgard, J. Harmatos, S. K. Mohalik, D. Roeland, and J. Niemöller, "Intent-driven Closed Loops for Autonomous Networks," *Journal of ICT Standardization*, pp. 257–290-257–290, 2021.

[2] J. Kaur, M. A. Khan, M. Iftikhar, M. Imran, and Q. E. U. Haq, "Machine learning techniques for 5G and beyond," *IEEE Access*, vol. 9, pp. 23472-23488, 2021.

[3] I. Aliyu, I. M. Abdullahi, S.-j. Lee, T.-W. Um, and J. Kim, "Towards Joint Optimization Problem for Computing and Resource Allocation in In-network Computing for Metaverse," 2022.

[4] *ITU-T Y.3000 series – Use cases for autonomous networks*, ITU-FGAN, 2022. [Online]. Available: https://handle.itu.int/11.1002/1000/15041

[5] *Architecture framework for Autonomous Networks*, ITU-FGAN, ITU-T Technical Specification, 2022. [Online]. Available: https://www.itu.int/en/ITU-T/focusgroups/an/Documents/Architecture-AN.pdf

[6] *Building a Digital Twin using Graph Neural Networks*, ITU-FGAN, 2021. [Online]. Available: https://extranet.itu.int/sites/itu-t/focusgroups/an/input/FGAN-I-058.pdf

[7] *Submission code github repository, team “Digital Twins”*, ITU-FGAN, 2021. [Online]. Available: https://github.com/FGAN-Digital-Twins/docker-network.git

[8] *Python microframework Flask, official website.*, Flask, 2022. [Online]. Available: https://flask.palletsprojects.com/en/2.2.x/

[9] ITU-FGAN. "Python WSGI HTTP Server for UNIX Gunicorn, official website." https://gunicorn.org (accessed Feb. 2, 2023).

[10] *TOSCA Simple Profile in YAML Version 1.3*, OASIS, 2020. [Online]. Available: https://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.3/TOSCA-Simple-Profile-YAML-v1.3.html

[11] *Report from Build-a-thon for ITU AI/ML in 5G Challenge 2022*, ITU-FGAN, 2022. [Online]. Available: https://extranet.itu.int/sites/itu-t/focusgroups/an/input/FGAN-I-239-R5.docx

[12] *xOpera orchestrator Github repository*, xOpera, 2023. [Online]. Available: https://github.com/xlab-si/xopera-opera

[13] S. Canny. "Python Docx " https://python-docx.readthedocs.io/en/latest/ (accessed 3/28, 2023).

[14] A. Roques. "PlantUML." https://plantuml.com/ (accessed 3/27, 2023).

[15] Neo4j. "Neo4j Graph database " https://neo4j.com/ (accessed.

[16] I. Delhi, ITU, and I.-E. ICT. "2022 Build-A-Thon: ITU India AI/ML Challenge Low Latency Closed Loop." IIT Delhi. https://bhartischool.iitd.ac.in/build_a_thon/index.html (accessed February, 13, 2023).

[17] Z. Xia, Z. Deng, B. Fang, Y. Yang, and F. Sun, "A review on sensory perception for dexterous robotic manipulation," *International Journal of Advanced Robotic Systems*, vol. 19, no. 2, p. 17298806221095974, 2022.

[18] A. R. See, J. A. G. Choco, and K. Chandramohan, "Touch, Texture and Haptic Feedback: A Review on How We Feel the World around Us," *Applied Sciences*, vol. 12, no. 9, p. 4686, 2022.

[19] Z. Zhou *et al.*, "A Sensory Soft Robotic Gripper Capable of Learning-Based Object Recognition and Force-Controlled Grasping," *IEEE Transactions on Automation Science and Engineering*, 2022.

[20] J. An, T. Li, G. Chen, Q. Jia, and J. Yu, "An Autonomous Grasping Control System Based on Visual Object Recognition and Tactile Perception," in *2022 International Conference on Service Robotics (ICoSR)*, 2022: IEEE, pp. 74-78.

[21] S. Gao, Y. Dai, and A. Nathan, "Tactile and vision perception for intelligent humanoids," *Advanced Intelligent Systems*, vol. 4, no. 2, p. 2100074, 2022.

[22] M. Kim, J. Yang, D. Kim, and D. Yun, "Soft tactile sensor to detect the slip of a Robotic hand," *Measurement*, vol. 200, p. 111615, 2022.

[23] N. W. Feed. "Nokia: 5G subscriptions in MEA to exceed 250 million by 2026." https://www.lightreading.com/5g/nokia-5g-subscriptions-in-mea-to-exceed-250-million-by-2026/d/d-id/779294 (accessed 03/06, 2023).

[24] Nokia. "Nokia report reveals three-fold increase in mobile data usage in India over the last five years." Nokia. https://www.nokia.com/about-us/news/releases/2023/02/16/nokia-report-reveals-three-fold-increase-in-mobile-data-usage-in-india-over-the-last-five-years/ (accessed 03/06, 2023).

[25] M. Hasan. "State of IoT 2022." https://iot-analytics.com/number-connected-iot-devices/ (accessed 03/06, 2023).

[26] R. Vaish and S. Matthews. "5G Will Accelerate a New Wave of IoT Applications." https://newsroom.ibm.com/5G-accelerate-IOT (accessed 03/06, 2023).

[27] 3GPP. "5G; Management and orchestration; 5G end to end Key Performance Indicators (KPI) (3GPP TS 28.554 version 16.7.0 Release 16)" ETSI. https://www.etsi.org/deliver/etsi_ts/128500_128599/128554/16.07.00_60/ts_128554v160700p.pdf (accessed 7/20, 2023).

[28] D. Li, D. Chen, J. Goh, and S.-k. Ng, "Anomaly detection with generative adversarial networks for multivariate time series," *arXiv preprint arXiv:1809.04758*, 2018.

[29] S. Ghosh and V. Kataria, "Multivariate time series unsupervised anomaly detection and diagnosis in 5g networks," 2020.

[30] D. T. Shipmon, J. M. Gurevitch, P. M. Piselli, and S. T. Edwards, "Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data," *arXiv preprint arXiv:1708.03665*, 2017.

[31] H. Ye, X. Ma, Q. Pan, H. Fang, H. Xiang, and T. Shao, "An adaptive approach for anomaly detector selection and fine-tuning in time series," in *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, 2019, pp. 1-7.

[32] L. F. Maimó, F. J. G. Clemente, M. G. Pérez, and G. M. Pérez, "On the performance of a deep learning-based anomaly detection system for 5G networks," in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2017: IEEE, pp. 1-8.

[33] C. Lian, H. Li, B. Zheng, T. Xu, and J. Han, "Anomaly Detection Modeling Based on Self-Adaptive Threshold Voting Integrating DBN-LRs," in *Proceedings of the 3rd International Conference on Big Data Research*, 2019, pp. 124-128.

[34] S. Nováczki, "An improved anomaly detection and diagnosis framework for mobile network operators," in *2013 9th international conference on the design of reliable communication networks (drcn)*, 2013: IEEE, pp. 234-241.

[35] J. Shi, G. He, and X. Liu, "Anomaly detection for key performance indicators through machine learning," in *2018 International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, 2018: IEEE, pp. 1-5.

[36] J. Burgueño, I. de-la-Bandera, J. Mendoza, D. Palacios, C. Morillas, and R. Barco, "Online anomaly detection system for mobile networks," *Sensors*, vol. 20, no. 24, p. 7232, 2020.

[37] *Framework for data handling to enable machine learning in future networks including IMT-2020* I.-T. Y.3174, 2020. [Online]. Available: https://www.itu.int/rec/T-REC-Y.3174-202002-I/en

[38] Cleveland, R.B., Cleveland, W.S., McRae, J.E. and Terpenning, I., 1990. STL: A seasonal-trend decomposition. *J. Off. Stat*, *6*(1), pp.3-73.

AUTHORS

Jaime Fúster De La Fuente received a B.Sc. in telecommunication technologies and a double M.Sc. in telecommunication engineering and cybersecurity from the ICAI Engineering School of Universidad Pontificia Comillas in 2019 and 2021, respectively. During his master's studies, he conducted research on the privacy and security of connected devices used by children and on Bluetooth low energy security within the framework of the EU H2020 project RAYUELA. As a research assistant at the Institute for Research in Technology of the ICAI Engineering School, he researched the security of Bluetooth and OBD-II technologies in vehicles. In 2021–2022, he received the "Vulcanus in Japan" scholarship and worked as an engineering intern at the Autonomous Networks Research and Innovation Department at Rakuten Mobile Inc. in Tokyo, Japan. His areas of interest revolve around privacy and security in distributed environments, decentralization, and privacy-preserving machine learning.

Álvaro Pendás Recondo received B.Sc. and M.Sc. degrees in telecommunication engineering from the Universidad de Oviedo, Gijón, Spain, in 2019 and 2021, respectively, where he is currently pursuing a PhD degree. Since 2019, he has been a research assistant with the Group of Signal Theory and Communications at the Universidad de Oviedo. He was an intern at the NASA Jet Propulsion Laboratory (JPL) Microwave Instrument Science Group, Pasadena, USA, in the summer of 2019; the German Aerospace Center (DLR) Institute of Communications and Navigation, Oberpfaffenhofen, Germany, in the summer of 2020; and the Rakuten Mobile Inc. Research and Innovation Department, Tokyo, Japan in 2022. His current research focuses on information theory and signal processing applied to multiuser wireless communications.

Paul Harvey is a lecturer in autonomous systems at the James Watt School of Engineering, University of Glasgow. He is exploring the cross-section of technologies and skills required for safe and meaningful fully-autonomous behaviour of both design and operation. Additionally, he is a working group co-chair in the UN's ITU-T Focus Group on autonomous networks and a visiting researcher at the University of Strathclyde. Paul also possesses extensive experience in industrial research, as he was the co-founder and research lead of the Rakuten Mobile Innovation Studio, and he is a strong believer in open and collaborative research.

Tarek Mohamed graduated from the Faculty of Engineering at Fayoum University, in the Electronics and Electrical Communications Department. His main interest is how we relate the telecommunications industry with different AI solutions on our way to having an AI-native network, and he is planning to attain a master's degree in this area.

Chandan Singh is an esteemed scientist with more than seven years of experience at TCS Research & Innovation Labs, where he focuses on pioneering research in the fields of robotics, machine vision, and AI. He has an M.Tech degree in electrical engineering from IIT Patna. Chandan's remarkable expertise lies in developing cutting-edge methods and algorithms to drive automation in various industries.

Vipul Sanap received his MTech in mechanical engineering from the Indian Institute of Technology (IIT) Jodhpur, India in 2020. He is currently a researcher at TCS Robotics Lab, TCS Research & Innovation, Delhi, India with active research interest in robotics systems and automation in the domain of warehouse and logistics and space robotics. With a decade of experience, he has contributed to the research and development of systems and frameworks for application of half humanoid to control crew pit panel assisting human for low orbit space exploration, and currently working on warehouse and logistics automation of material handling systems. Specifically, his current research interests encompass dexterous manipulation of robotic grasping systems, computer vision techniques and machine-learning algorithms to enable robots to perform tasks autonomously.

Ayush Kumar completed his B.Tech in electrical engineering from the Indian Institute of Technology, Patna in 2018. Currently, he is working as a researcher in the TCS Research & Innovation Labs. His major research interest lies in using Robotics, warehouse automation, and autonomous mobile robots.

Sathish Venkateswaran is a System Architect, Analytics & AI/ML, NM Architecture R&D, Nokia Bengaluru with 18+ years in the Telecom industry having experience in SW development of Network Management for Radio Networks. He is Interested in learning and applying aspects of technology specifically in the area of AI/ML & Analytics for Telecom networks

Sarvasuddi Balaganesh is a software Engineer, NM R&D Nokia Bangalore with 3+ years in the Telecom industry having experience in SW development of Network Management for Radio Networks. He is Interested in learning and exploring in the area of Machine Learning for Telecom networks.

Rajat Duggal has 22+ years in the telecommunications industry (2G/3G/4G/5G) with experience of field experience in radio access, transport, E2E network development, deployment/configuration experience and he has also worked as a 5G R&D architect in radio networks. He is interested in discussing and learning aspects of technology both with the industry and academia (which is the source of the talent pool to cater to the industry). He has contributed in 5G patents.

Sree Ganesh Lalitaditya Divakarla is affiliated with PES College, India, Bengaluru.

Vaibhava Krishna Devulapali is affiliated with PES College, India, Bengaluru.

Ebeledike Frank Chukwubuikem is an undergraduate student of computer engineering at the Federal University of Technology Minna, Nigeria. He is also an active member of the Microsoft ADC Student League (MASL) FUTMinna Chapter. He has research interests in AI for autonomous networks, 5G and blockchain technology.

Emmanuel Othniel Eggah is an undergraduate student of computer engineering at the Federal University of Technology Minna, Nigeria. He has research interests in ML and AI for autonomous networks.

Abel Oche Moses received his diploma in word processing and desktop publishing from World Link Computers, Benue State, Nigeria, in 2014. He also attained a National Diploma in Computer Engineering from Benue State Polytechnic, Ugbokolo, Nigeria, in 2017, and he is currently undergoing a B.Eng program at the Federal University of Technology Minna, Nigeria. He is a member of TechRanger, which took part in the ITU Focus Group on Autonomous Networks (Close Loop) Challenge, in 2022, and now he is a mentor of the WINEST Research group.

Nuhu Kontagora Bello is a lecturer in the Department of Computer Engineering at the Federal University of Technology Minna, Nigeria. He obtained his M. Tech in computer science and Engineering at the Ladoke Akintola University of Technology Ogbomoso, Nigeria. Moreover, he attained a B.Eng in electrical & computer engineering from the Federal University of Technology Minna, Nigeria in 2010. He is currently a doctoral student at Ahmadu Bello University, Zaria, Nigeria. His research interests include artificial and computational intelligence, localization in sensor networks, computer/network security, Internet of Things (IoT) and software-defined networking.

James Agajo received a Bachelor of Engineering in electrical and computer engineering from the Federal University of Technology Minna, a Master's of Engineering in electronics and telecommunication engineering from Nnamdi Azikiwe University, and a PhD in telecommunication and computer engineering from Nnamdi Azikiwe University. A former HOD, acting director and postgraduate coordinator, he is currently an associate professor and the head of the Department of Computer Engineering with the Federal University of Technology Minna, School of Electrical Engineering and Technology. He has published over 130 articles and has received many awards, including the IBM AI Analyst Master's Award, the IBM AI Analyst Award, the IBM IoT Cloud Developer Award, the IBM Blockchain Developer Award, and the IBM Telecommunications Insights & Solutions.

Wael Alron is a telecommunications implementation and planning and subject matter expert with 14+ years of experience. He leads ICT cross-functional projects coordinating with global partners, vendors, consultants and customers and ICT contractors. He interfaces with standardization and guidelines with telecommunication industry players. He participates in events of the global Telecom bodies GSMA, GCF and ITU. He is at an expert level in project implementation and product management, alongside commercial business initiatives. He is leading the standardization and guidelines involvement as a technology standard for IBS technologies such as GSM, GPON and FTTX and telecommunication structural cabling and network infrastructure. He started his career at Trend Technologies for GPON projects in Dubai and then moved to EMIRATES Telecommunications Group Etisalat for managing ICT contractors to apply guidelines and new technologies and acceptance of the projects. He worked with Datwyler ME& Swiss as a technical manager for ICT manufacturing and engineering solutions. He then moved to Alcatel-lucent (Nokia) as a deployment manager of telecommunications in the Middle East to lead an in-house team and vendors for ICT projects handled and accepted by regional operators. The cooperation involved creating new initiatives of advanced technologies and highly supporting the other departments to achieve targeted objectives and strategies. He currently works with Du in UAE.

Fathi Abdeldayem is a telecommunications standardization expert with 25+ years of experience. He leads cross-functional projects coordinating with global key partners, vendors and customers. He interfaces the cooperation from a standardization point of view with global Tier 1 telecommunication industry players and global standardization bodies of GSMA, GCF, ITU, ETSI, 3GPP, LCA and the MetaVerse Standardization Forum. He is at an expert level in project management and product management, as well as international business management. He leads the technology standardization involvement as a technology standards expert interfacing with GSMA, GCF, ITU, ETIS, 3GPP, LCA and the MetaVerse Standardization Forum. He started his career at Siemens in Germany as an R&D MSC, a student in Chipset and terminal development. He then moved to the automotive industry working for BMW and Audi

(Germany) and then returned to the telecommunications branch as a consultant expert leading an IOT with O2 Germany. He managed different accounts handling the technology division, business division, and the after sales and customer care division. The cooperation involved creating a strategy and roadmap for the company's terminal portfolio. He then worked at Telefonica Germany included developing and managing E2E of the terminal portfolio for Telefonica Germany end users including software engineering, requirements engineering, and managing Tier 1 key terminal vendors such as Nokia and Apple. He worked in an expert position launching technology services such as LTE, VoLTE, VoWiFi,5G NSA -SA, e-SIM Smartphone OS expert level, test systems smart city, Internet of Things, AI, ML, LiFi standardization, and MetaVerse standardization. He won the 1st place Innovation Prize from Du during the H H Sheikh Mohammed bin Rashid Al Maktoum Innovation Week. Additionally, he is the founder of the BE-AT-The Top work process, the initiator of the UAE Profile Initiative and the initiator of the 5G MENA Conference in cooperation with the Global Certification Forum.

Melanie Espinoza Hernández is a student of electronic engineering at the TEC of Costa Rica and has a professional technical degree in software development. Since 2019, she has been part of the Descubre Robotica program. She is the director of technological strategies and is developing several personal projects based on STEM learning. Her main project is called the Internet of Bees, and it includes the modernization of beekeeping using various elements of Industry 4.0, such as artificial intelligence applied to the monitoring of intensive systems.

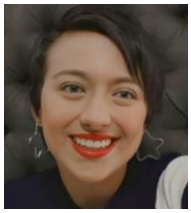
Abigail Morales Retana (team leader) received a B.S. in electromechanical and electrical engineering from Universidad Fidélitas, San José, Costa Rica, in 2023. Now, she continues her postgraduate studies in the same field. She participates in a project called ARCI, an Autonomous Smart Delivery Fleet, which is a challenge that raises the importance of artificial intelligence for the collaborative service of different automated equipment in future smart cities. She has also been an assistant for the mentoring program at Descubre Robótica since 2019.

Jackeline García Alvarado will finish her studies at the United World Colleges (UWC) in 2023 for an International Baccalaureate Diploma Program (IBDP). Her subjects of interest include Biology, Spanish Language and Literature, English B, Environmental Systems and Societies, Visual Arts and Math Applications and Interpretations. She is a gender champion, a regional representative of the Americas, and was one of the 12 selected as a gender champion to attend the International Telecommunication Union Global Youth Summit in Rwanda, Africa. She was also selected by the United Nations International Telecommunications Union to represent Generation Connect Youth Envoys at the Plenipotentiary Conference in Romania (PP-22).

Nicolle Gamboa Mena has a professional technical degree in software development and certified training for AWS cloud skills. She has been part of Costa Rican service companies with cloud technology solutions and is currently studying industrial maintenance engineering. She has been part of the Descubre Robótica program since 2019, and actively fulfils the role of the digital transformation director, driven by Industry 4.0, for the development of the Bee Code initiative of the proposal for the digitalization of native bees: Digital Bees.

Juliana Morales Alvarado is affiliated with Descubre Robótica, Costa Rica.

Ericka Pérez Chinchilla (team leader) received a B.S. degree in systems engineering from Universidad Fidélitas, San José, Costa Rica, in 2021. She is continuing her postgraduate studies as a student of technology resources, computer and information sciences and support services. With her experience in mentoring disciplines for other young women, she has dabbled in STEM issues for women. She has also been an assistant for the mentoring program at Descubre Robótica since 2022.



Amanda Calderón Campos is a student of mechatronics engineering at the TEC of Costa Rica and pure physics at the University of Costa Rica. In addition, she is a graduate of the System of Scientific Colleges. She has been part of the Descubre Robótica program since 2021 and has been actively involved in astrophysics and aerospace issues. In addition, she has led projects that evaluate alternatives for the use of green hydrogen ecosystems, their production and their respective metrics.

Derek Rodríguez Villalobos is affiliated with Descubre Robótica, Costa Rica.

Oscar Castillo Brenes (mentor) is the chief executive and strategic planner of Technomakro, an international consulting firm, with more than 25+ years of experience in automation and industrial controls, electronics, IT infrastructure, geospatial data and scientific robotics supporting computer network applications and services. In turn, he fulfilled the role of senior collaborator and leader, leading projects and engineering teams and focusing on providing business value to meet business objectives. He is certified in quality control systems, as an ISO 9000, 14000, and 18001 auditor. He is co-founder of the Descubre Robótica program, which is a fraternity that specializes in providing student and teacher training to prepare them in the latest areas of knowledge (robotics, computer vision and machine learning and applied science to solve complex problems) to pave the way for the generation within the fourth industrial revolution.

Kodandram Ranganath is affiliated with Nokia Networks Bengaluru, India.

Ayushi Khandal is affiliated with Nokia Networks Bengaluru, India.

Rakshesh P Bhatt is affiliated with Nokia Networks Bengaluru, India.

Kunal Mahajan is affiliated with RVCE, Bengaluru, India.

Prikshit CS is affiliated with RVCE, Bengaluru, India.

Ashok Kamaraj is affiliated with Nokia Networks Bengaluru, India.

Srinwaynti Samaddar is affiliated with Nokia Networks Bengaluru, India.

Sivaramakrishnan Swaminathan is affiliated with Nokia Networks Bengaluru, India.

M Sri Bhuvan is affiliated with RVCE DSCE, Bengaluru, India.

Nagaswaroop S N is affiliated with RVCE DSCE, Bengaluru, India.

Blessed Guda is a graduate student at Carnegie Mellon University, Africa. He made several contributions to the ITU Focus Group on ML for 5G and autonomous networks. He has research interests in AI for NLP, network security, 5G and autonomous networks and embedded systems. He is a mentor with the WINEST Research Group and founder of the AI4Africa Research Group. He has won awards for his participation in ITU AI challenges in AN and TinyML.

Ibrahim Aliyu received his PhD in computer science and engineering from Chonnam National University, South Korea, in 2022. He also attained a B.Eng and M.Eng in computer engineering from the Federal University of Technology, Minna, Nigeria, in 2014 and 2018, respectively. He is currently a postdoc researcher at the Hyper Intelligence Media Network Platform Lab, Dept. of ICT Convergence System Engineering, Chonnam National University. In addition, he is contributing to the ITU Focus Group on autonomous networks. His current research interest is on source routing-based in-network computing for XR/metaverse applications and the development of a zone adaptive network structure for large-scale metaverse deployment. His other research interests include federated learning, data privacy, network security and AI for autonomous networks. He received the Mentors Encouragement Award from ITU AI/ML in the 5G Challenge in 2021 and the 2017 Korean Government Scholarship Program Award.

Jinsul Kim received a B.S. in computer science from the University of Utah, Salt Lake City, Utah, USA, in 1998, and an M.S. and PhD in digital media engineering, Dept. of Information and Communications from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2004 and 2008.

He worked as a researcher in the IPTV Infrastructure Technology Research Laboratory, Broadcasting/Telecommunications Convergence Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea from 2004 to 2009. He worked as a professor at Korea Nazarene University, Cheonan, Korea from 2009 to 2011. Currently, he is a professor at Chonnam National University, Gwangju, Korea. He has been a reviewer for IEEE Trans. Multimedia since 2008 as an IEEE Member, and he has been invited to the TPC (Technical Program Committee) for IWITMA2009/2010, PC (Program Chair) for ICCCT2011, IWMWT2013/2014/2015 and general chair for ICMWT2014. His research interests include QoS/QoE, cloud computing, edge computing, AI, energy AI, multimedia communication and new media (VR, AR, metaverse etc.)

Vishnu Ram OV has 24 years of hands-on experience in the telecommunications industry, developing and implementing standards, holding 13 internationally granted patents, and has published many papers and was appointed as a Scientific Advisory Board Associate (SABA) member of Motorola Networks. He is currently serving as an independent consultant, vice chair of the ITU-T Focus Group on Autonomous Networks (ITU-T FG-AN), and was co-editor of the recently published ITU-T focus group ML5G specifications on AI/ML, which led to many Recommendations such as ITU-T Y. 3172. He is a senior member of IEEE. His current passion includes coordinating global standards in ITU-T, liaison with other SDOs like ETSI and IRTF, mentoring student projects, and coordinating global challenges in AI/ML in 5G.