Collaborative Cross System AI: Towards 5G System and Beyond

Miloud Bagaa, Tarik Taleb, Jukka Riekki, and JaeSeung Song

***Abstract*—The emerging industrial verticals set new challenges for the 5G and beyond systems. Indeed, the heterogeneity of the underlying technologies and the challenging and conflicting requirements of the verticals make the orchestration and management of networks complicated and challenging. The recent advances in network automation and artificial intelligence (AI) create enthusiasm from industries and academia towards applying these concepts and techniques to tackle these challenges. With these techniques, the network can be autonomously optimized and configured. This paper suggests a collaborative cross-system AI that leverages diverse data from different segments involved in the end-to-end communication of a service, diverse AI techniques, and diverse network automation tools to create a self-optimized and self-orchestrated network that can adapt according to the network state. We align the proposed framework with the ongoing network standardization.**

***Index Terms*—5G, Beyond 5G, Cloud Computing, Edge Computing, and AI.**

I. Introduction

5G and beyond systems are not only about increasing the network throughput, but also target a large number of services and applications that will transform our daily lives. In the future, technologies such as augmented reality, virtual reality, and holographic telepresence will be commonly used, enabling new applications, such as industry 5.0, industrial internet of things (IIoT), and self-driving cars. The network will adapt itself using various technologies to provide more extensive broadband, near-instant, efficient, resilient, and reliable connectivity with five-nines network availability. This development sets challenges to orchestrating and managing the network. The envisioned network will be complex and heterogeneous and will consist of various technologies and industrial verticals with high key performance indicators (KPIs). The network must adapt to tackle these challenges and achieve the objectives by providing an intelligent interplay between edge and cloud resources.

Fortunately, artificial intelligence (AI) has been matured enough to provide efficient solutions for complex problems, including automation industries, telecommunications, and transportation. Moreover, the emerging advancements in computation and networking enable AI techniques for managing the complexity and heterogeneity in the 5G and beyond systems [1]. Machine learning (ML), a particular case of AI, will play a crucial role in the future network and industrial verticals. ML techniques provide the ability to learn from the environment and enable self-optimized systems that adapt according to the states of the network and the industrial verticals. Furthermore, smooth service management and orchestration can be achieved with intelligent agents, from distributed artificial intelligence (DAI). Such agents perceive their environment and act to achieve a common goal. Many DAI learning techniques have been widely used in the literature, including transfer learning, federated learning (FL), meta-learning framework (MLF), and multi-agent reinforcement learning (MARL) for ensuring fluent collaboration between agents. FL enables the distribution of the learning process among multiple agents, while a central server manages the global model. MLF aims to overcome the limitations of FL by enabling more extensive learning processes and more complex tasks. Meanwhile, MARL offers the agents (e.g., self-driving cars) with the possibility to interact among themselves and also with the environment, and to learn how to optimize the distributed decisions taken by different agents.

This paper leverages DAI techniques and the ETSI Zero-touch network and Service Management framework[1], enabling prompt and effective, combined management and orchestration in an autonomous and blended way. We propose a collaborative cross-system AI that manages resource heterogeneity and orchestration complexity, as well as the geographical distribution of clouds and edges. The proposed framework leverages and enhances DAI techniques to deal with the above-mentioned challenges and to meet the desired objectives and KPIs targeted by the emerging industrial verticals and applications. The framework's output is a network intelligence function (NIF) integrated into different management domains for enabling smart self-orchestration and self-management of end-to-end network slices. This paper illustrates how the NIFs leverage AI techniques, such as DAI, FL, MLF, and MARL, to build a collaborative system that can efficiently make the right decisions. The NIF agents are integrated into the management domains from the industrial verticals to the core network. We have designed our framework to be aligned with the ongoing standardization, including ETSI MEC [2] and 5G service-based architecture (3GPP TS 29.520 (v15.1.0)). We also show

Miloud Bagaa and Tarik Taleb are with the Department of Communications and Networking, School of Electrical Engineering, Aalto University, Finland (e-mails: {firstname.lastname}@aalto.fi). Jukka Riekki is with the Faculty of Information Technology and Electrical Engineering, Oulu University, Finland (e-mail: jukka.riekki@oulu.fi). Tarik Taleb is also with the Faculty of Information Technology and Electrical Engineering, Oulu University, and with the Department of Computer and Information Security, Sejong University, Seoul 05006, South Korea.

JaeSeung Song is with the Department of Computer and Information Security, Convergence Engineering for Intelligent Drone, Sejong University, Seoul 05006, South Korea (Email: jssong@sejong.ac.kr).

[1]ETSI GS ZSM 002, "Zero-touch network and Service Management (ZSM): Reference Architecture"

how NIF agents, provided within the collaborative cross-system AI framework, benefit a wide range of applications that vary from dynamic network customization to autonomous cars, object recognition, and smart home management.

The rest of the paper is organized as follows. Section II summarizes different AI technique enablers and different network automation and orchestration frameworks. In Section III, we present the collaborative cross-system AI, NIF agents and their sources of information. Section IV describes use-cases whereby NIF agents can be used for unified smart orchestration and management. Finally, Section V concludes the paper.

II. Related Work and Background on Cross-System AI enablers

A. Machine Learning Techniques

ML has recently gained considerable attention from academia and industry thanks to its ability to provide smart and scalable solutions. ML techniques, such as supervised learning, unsupervised learning, and reinforcement learning, have been widely adopted by network, cloud, and telecom providers. With these techniques, a network can learn from the environment and realize self-optimization and self-configuration for ensuring the desired KPIs. These KPIs are continuously and dynamically changing due to heterogeneity in vertical demands, mobility patterns, and dynamic workload. Moreover, these techniques can provide fast settings while making an abstraction on the environment, facilitating the integration of the same solution into different scenarios. Hereby, abstracting the complex environment decreases the number of assumptions in the modeling, thus improving the accuracy.

Supervised Learning: In supervised learning algorithms, the data's inner relations may not be known, but the model's output is known. This technique requires labeled training data that is used for training the model. The latter is mainly used for either classification or regression problems. Training requires three sets of data. The most important part of the data is used for training, while the two remaining pieces are used for testing and evaluation.

Unsupervised Learning: Unlike the supervised learning approach, the data is unlabeled in these techniques. Relevant models try to find a correlation between the input data and classify it into different clusters (i.e., clustering).

Reinforcement Learning: The reinforcement learning (RL) technique has been widely used in the literature for self-optimizing a continuously and dynamically changing network. Reinforcement learning belongs to the same family as the Markov Decision Process (MDP). MDP is a model-based approach (i.e., Transition probability), whereas RL is a model-free approach. RL adopts a unique model training method that is based on trial and error and reward functions. The RL agent periodically makes decisions, observes the environment, and then adjusts the next action policies for achieving optimal configuration. RL summarizes the environment and actions in a Q-table. Unfortunately, the Q-table cannot provide an optimal strategy in a complicated situation with many states and actions. A new paradigm, dubbed deep reinforcement learning (DRL), has been recently suggested to overcome the limitations of RL. DRL leverages Deep Learning (DL) [3] for presenting the Q-table as a function by leveraging the strength of neural networks. DRL is mainly classified into the following classes: i) Value-based approach, such as Deep Q Network (DQN), Double DQN (DDQN), and Dueling DQN (Duel-DDQN) Algorithms; ii) Policy-based approach, such as REINFORCE (Monte-Carlo Policy Gradient) and REINFORCE with baseline; iii) Actor-Critic approach, which is a hybrid between the two previous approaches. Many algorithms that use Actor-Critic have been suggested in the literature, such as Advantage Actor Critic (A2C), Asynchronous Advantage Actor Critic (A3C), Deep Deterministic Policy Gradient (DDPG), and Proximal Policy Optimization (PPO) Algorithms [4].

The objective of ML algorithms is mainly to learn parameters for generating a function F that maps the input data (i.e., features) to an output (i.e., predictions). In the case of deep learning (DL), which is a particular case of ML, the parameters can be neural weights, bias, and batch normalization parameters. Formally, ML adopts a data-driven approach by optimizing the parameters by mapping the changes between the input and output data. Different gradient descent-based optimization techniques are adopted in the literature, such as gradient descent with momentum (GDM), stochastic gradient descent (SGD), RMSprop, and Adaptive moment estimation (Adam). The latter leverages both GDM and RMSprop for achieving better performances in terms of variance and bias tradeoff.

Besides the gradient descent optimizer and the learning rate, other methods have been adopted in DL to speed up the convergence and prevent the vanishing and exploding gradient problems, such as normalization (i.g., batch normalization), regularization (e.g., dropout, Frobenius norm, and Xavier initializer). Various activation functions have been also adopted to break the linearity between successive neural network layers, such as Sigmoid, Tanh, and ReLU. Without these activation functions, whatever the number of neural layers, the deep neural network can be presented as a simple linear or logistic regression function. The underlying ML algorithm uses a dataset containing instances with their respective classes (in case of supervised learning) to derive the function F or look for a correlation between attributes to regroup them into similar categories (unsupervised learning). Most of the applications are made locally, whereby a data source is used to feed the learning component holding the model. The model's success hinges on the learning algorithm, the attributes selection, and, more importantly, the data source. The latter plays a significant role: according to its volume and diversity, it is possible to extract exciting behaviors that are hard to discover with regular data analysis.

B. Distributed Artificial Intelligence and Multi-Agent Systems

Distributed Artificial Intelligence (DAI) is seen as a subfield of Artificial Intelligence. It has attracted a colossal attention due to its capability for solving complex problems. It solves problems related to coordination, concurrency, and decision-making. DAI systems are composed of agents, which are computational entities, software programs, or robots that perceive their environment, make decisions based on the perceptions and act accordingly, thus changing the state of the environment. Traditional AI models focus on solving problems by searching through a space of potential solutions. DAI can be classified into three different categories according to the underlying used methods: i) Parallel AI; ii) Distributed Problem Solving (DPS); and iii) Multi-Agent Systems (MAS).

Multi-Parallel AI has been proposed to enhance the efficiency of legacy centralized AI algorithms. It enables the development and employment of parallel AI Algorithms using different languages and architectures. Numerous software and tools have been adopted in the literature for promoting Multi-Parallel AI systems, such as Spark, GraphLab, and TensorFlow. From another side, DPS divides a task into subtasks and then allocates them to a set of AI cooperative computing entities. The latter communicate among themselves and share the knowledge and resources to perform the final task. Unfortunately, DPS suffers from the lack of flexibility and automation. A MAS is a complex system that consists of several agents designed to carry out specific tasks. These agents cooperate among themselves to achieve a common objective. The participating agents and their coordination are usually defined at design phase to accomplish the task. In contrast to the previous two classes, a MAS agent has the ability to learn and make autonomous decisions and actions. A MAS agent is able to autonomously interact with other MAS agents and with the environment to learn new contexts and actions. The MAS agents can be divided into two categories:

Active Agents: do not only learn from their surroundings but can also impact their respective environments.

Passive Agents: act as passive watchers, acquiring knowledge from their respective environments without necessarily affecting them.

C. Distributed Federated Learning and Multi-Agent Reinforcement Learning

1) Federated Learning: Federated learning (FL) is a ML technique that enables the distribution of the learning process among many parties (e.g., mobile devices, edges, or clouds). These parties collaborate with a central orchestration server (e.g., service provider) while keeping the training data decentralized. In this ML technique, the learning agents can collaboratively train a global learning model without sharing their local datasets. The central agent will learn from the different agents and accordingly help the other agents for enhancing their local models, thus perceiving improved learning performance. Different techniques can be leveraged for enabling DLF, such as transfer learning by either initializing the parameters or freezing the hidden layers. The DFL mechanism has the following characteristic features: i) Decentralized computation that leads to achieving the optimal learning rate in an optimal time; ii) Sharing the gained knowledge (i.e., trained models) with the new members; iii) Increasing data privacy (i.e., images and videos) by keeping the generated data close to their source origination.

Authors in [5] suggest MOCHA, which is a framework for federated multi-task learning. MOCHA aims at reducing communication costs and ensuring fault tolerance. Meanwhile, authors in [6] have proposed supervised federated learning, dubbed FedProx, that aims at optimizing the local model to fit the local datasets. In contrast, the global model is optimized to perform well on distributed datasets by aggregating the local learning parameters. However, such conventional federated learning is mainly focused on a single learning task (i.e., the global model) with non-i.i.d (non-independent and non-identically distributed) datasets. Moreover, the global model can be biased by agents that have massive datasets and generate a large number of updates [6]. On the other hand, the generalization and specialization of sub-models can hinder the convergence (i.e., overfitting or underfitting) of the global model. To overcome the heterogeneity of the underlying data and its distribution at different agents, authors in [7] suggest a Model-Agnostic Meta-Learning (MAML) framework that leverages federated averaging technique for providing a more personalized model for each agent and hence offers better model convergence. Unfortunately, the MAML framework lacks cohesion relations between the generalization and the personalization.

Regrettably, the DFL technique comes with unavoidable communication overhead and complexity that impair the applicability of this technique. An extra communication overhead between the agents is required for ensuring a coherent vision of the global model. Moreover, while the splitting of the dataset has a positive impact on data privacy, the split of the dataset on multi-agents could harm model accuracy and precision and the tradeoff between the bias and variance in the ”global” model.

2) Meta Learning Framework - MLF: FL leverages a distributed multi-agent environment for learning a model for a task using multiple datasets. However, the resulting global model is adequate to address only this task due to its rigid design, limiting the practicality of the model in complex scenarios or unseen data. To mitigate these issues, MLF [8] has been proposed to enable generalization from a broad training data of similar tasks. The model provided by MLF is designed to be easy to fine-tune by modifying the gradient descent method, hence enabling generalization in the prediction.

3) Multi-Agent Reinforcement Learning: In the literature, the model-based approach, i.e., Markov Decision Process (MDP), has been extended from a single agent to a multi-agent system by forming a stochastic game that enables extensive and exciting use-cases. The model-free approach, represented by reinforcement learning (RL) techniques, has recently gained a lot of momentum. A multi-agent system employing RL is called multi-agent reinforcement learning (MARL) [9]. This approach introduces benefits for distributed systems, such as autonomous driving and network self-management and self-orchestration, where more than one agent should collaborate to achieve the desired objectives. The system state and reward are split among the agents for achieving a common goal.

MARL algorithms can be classified into three main approaches: i) fully cooperative; ii) fully competitive; and iii)

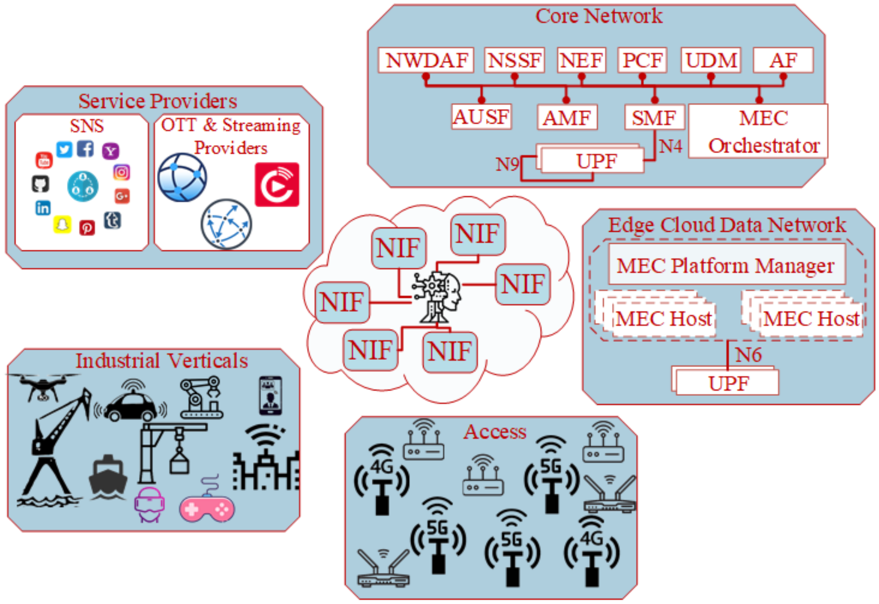


Fig. 1: Collaborative Cross System AI – High-Level Architecture.

hybrid. While the agents collaborate to optimize a single objective in the first approach, the second approach's agents compete to enhance their benefits, similar to a zero-sum game where mixed-strategy Nash equilibrium is sought. In the third approach, the first two approaches are mixed to optimize both the global objective and each agent's goals. This approach is similar to the general-sum game, where the dominant strategy and Nash equilibrium are sought in pure- and mixed-strategies. MARL leverages theories varying from optimization theory, dynamic programming, and game theory to decentralized control.

D. Network Automation and Cognitive Network

ETSI established the Zero-touch network and Service Management Industry Specification Group (ZSM ISG) that enables agile, efficient unified management and orchestration in an autonomous and harmonized way. This concept mitigates the complexity of the next-generation networks. The ZSM framework is envisaged as a next-generation management system that aims at having all operational processes and tasks (e.g., planning and design, delivery, deployment, provisioning, monitoring, and optimization) executed automatically. Moreover, the framework ensures both scalability and resiliency by adopting a decentralized approach and giving more freedom to the local sub-system management for making local decisions. ZSM consists of loosely coupled services with self-management capability by leveraging the flexibility of SDN/NFV technologies and the smartness of AI/ML to ensure a full-automation system that reduces human intervention. The ZSM architecture is designed in a modular way that consists of self-contained and loosely coupled services that communicate using intent-based interfaces. The latter exposes high-level abstraction to hide the complexity, technology, and vendor-specific details. Flexibility and extensibility are improved as new services (horizontal/vertical control scaling) and capabilities can be added to different sub-systems. The ZSM architecture consists of two management parts, namely $i)$ management domains and $ii)$ E2E service management domain.

Similar to the ETSI ZSM initiative, many research works have been proposed in the literature towards enabling cognitive and self-managed networks. The authors in [10] propose a learning augmented optimization approach that leverages deep learning techniques and Lyapunov stability theories for managing different network slices. Both historical records and real-time observations are used for controlling network slices. To reduce human intervention in the network and associated errors, authors in [11] propose a framework that provides intelligence and autonomy to systems by leveraging AI/ML methods. Wang et al. [12] propose a cognitive network-oriented framework to provide suitable Quality of Experience (QoE) using ML techniques in an SDN architecture. Application management strategy can be learnt based on information obtained from the South Bound Interface (SBI), whereas network metrics and KPIs achieved are monitored based on information retrieved from the North Bound Interface (NBI). Sciancalepore et al. [13] proposed a zero-touch orchestration (z-TORCH) approach to optimize the Quality-of-Decisions (QoD). The framework leverages both unsupervised learning and reinforcement learning for placing VNFs at minimum monitoring costs. Authors in [14] proposed a DeepCog framework that forecasts the needed resources for accommodating network traffic while preventing the over-provisioning of resources and ensuring the desired resource requests. The framework is based on deep learning technique for achieving the desired objectives.

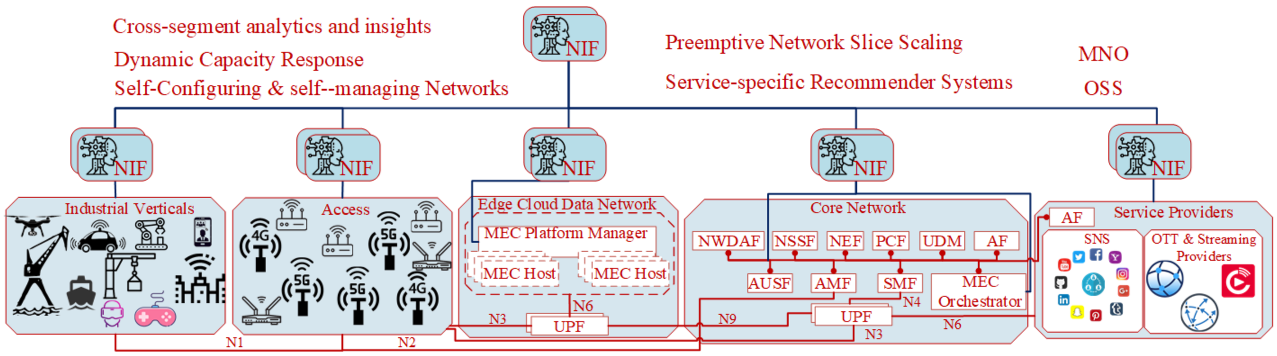


Fig. 2: Collaborative Cross System AI – Conceptual Architecture.

III. Cross System AI for Enabling Smart Next Network Generation

A. Cross System AI Framework – Main Overview

Fig. 1 depicts an overview of the proposed framework for enabling collaborative cross-system AI. Network Intelligence Functions (NIFs) are integrated into the management domains for enabling autonomous management and supporting collaboration amongst these domains. A NIF agent is designed to produce decision making insights and recommendations based on information gathered from various management domains (i.e., representing the different segments involved in an end-to-end communication path of a service). NIF agents offer the agility and autonomy in network and service management required by the 5G and beyond systems. We have designed our framework to run separate NIF agents by each management domain to ensure service granularity, scalability, availability, flexibility, and load balancing.

The NIF agents communicate through the domain integration fabric for ensuring smooth collaboration between the management domains. The collaboration and self-optimization of the NIF agents are based on the AI enablers, including ML, DAI, MAS, FL, MLF, and MARL. The insights and recommendations generated by the NIF agents are then delivered to the respective network functions (NFs) to accordingly adapt their behavior. In this context, the NFs benefit from the NIF agents as they are able to make intelligent decisions that suit different verticals from the service provider to the user equipment (UE).

As depicted in Fig. 2, our framework consists of two hierarchical levels. While the first level consists of the end-to-end management domain, the second level consists of five management domains. The management domains are merged in a unified manner, based on the 5G architecture defined by 3GPP and the MEC specifications. AS depicted in the figure, the different management domains are interconnected.

End-to-end management domain: is responsible for orchestrating the whole system for ensuring the desired KPIs and service level agreements (SLAs) targeted by different verticals. This management domain is formed by multiple NIF agents that may run on top of cloud-native environments and collaborate to ensure the above-mentioned objectives.

Management domains: are self-organized and self-managed to ensure the end-to-end KPIs and SLAs. In each management domain, the NIF agents, besides their internal collaborations, also collaborate with other NFs in that management domain to ensure the desired objectives. Based on the different segments that are typically involved in the end-to-end communication path of a service, we distinguish five main management domains: $i)$ industrial verticals domain; $ii)$ access domain; $iii)$ edge cloud and data network domain; $iv)$ core network domain; and $v)$ service provider domain.

B. Sources of Information at Management Domains

The focus of this work is on mobile services consumed by different types of end-users. Various stakeholders are involved in this ecosystem that forms the entire service-delivery system. Each stakeholder owns (or rents) and administrates a sub-system (i.e., management domain). Gathering analytical data across different mobile network segments can provide valuable insights into the optimization and automation of mobile network operations. The network segments vary from the end-user devices to the service provider crossing fronthaul, backhaul, and core network. The gathered analytical data would have a significant impact on the network softwarization era, whereby software-defined networking (SDN), networks function virtualization (NFV), and MEC technologies will play a crucial role in 5G and beyond mobile systems. Hereunder, and for the sake of clarity through concrete examples, we cover some of these information sources, classified as per the respective sub-system.

1) Industrial Verticals: The user equipment holds data that can be used to improve the network by predicting the behavior of a mass of users. NIF agents will leverage these data for making decisions. This data varies in type and how it can be collected according to the type of UE. We present the following non-exhaustive list:

User Information: Characteristics of users that do not reveal personal information.

Service Usage Information: Installed and running applications, localization history, perceived QoE measurement (through response time, throughput, delay), data consumption history, etc.

Device-specific Information: Information related to the device and operating system, energy consumption, usage activity, connected accessories, etc.

The end-devices' generated data is leveraged by different NIF agents to predict the behaviors and accordingly allocate the required network and computation resources. The data is generated and received as time-series data by the NIF agents. For this reason, NIF agents apply different data preprocessing techniques, such as numerosity reduction, and data cleaning, compression, and reduction. For instance, the data reduction can be managed using discrete wavelet transform (DWT) and discrete Fourier transform (DFT) techniques.

2) Radio Access Network: Different AI enablers can be used to generate insights from the information gathered from RAN. These insights help NIF agents deployed at the access network to predict the resource blocks utilization and overheard, as well as the mobility and positions of various users and devices. Then, accordingly, network and computational resources would be relocated from an access anchor to another in order to ensure the desired KPIs and SLAs.

3) Multi-Access Edge Cloud: According to ETSI's white paper ”MEC in 5G” [2], the actual means of communication in MEC follows the approach of previous NFs by implementing a RESTful API. The MEC infrastructure is divided into two levels: a System Level which holds the MEC Orchestrator, and a Distributed Host Level that consists of MEC hosts. The MEC orchestrator communicates with the core network mainly through the Network Exposure Function (NEF). It has also the ability to establish communications directly to NFs if proper authorizations are set. The MEC orchestrator has the following capabilities:

- Local routing and traffic steering,
- Influence on UPF (re)selection and traffic routing (via Policy Control Function - PCF or NEF), and
- Session and Service Continuity (SSC) modes for different UEs and application mobility.

As depicted in Fig. 2, NFs at each MEC hosts are connected to user plane functions (UPFs) located at the MEC hosts using $N6$ interfaces. Using ML, DAI, MAS, FL, MLF, and MARL approaches, NIF agents at the end-to-end management domain and MEC management domains collaborate to place and assign adequate resources to different NFs for ensuring the desired KPIs and SLAs. Following the specifications defined by ETSI regarding the MEC orchestration[2], the information leveraged by NIF agents located at different MEC management domains are summarized in Table I.

4) Core Network: In the 5G architecture, 3GPP defined a function called Network Data Analytics Function (NWDAF). The goal of NWDAF is to provide analytic information related to the network. According to 3GPP TS 29.520 (v15.1.0) this function offers two kinds of operations. The first one, subscriptions, lets other NFs subscribe to events for a defined network slice instance or all network slice instances, and a threshold can also be specified. NWDAF will then send notifications to the respective NFs when the event occurs. The second operation, analytics, lets the NFs directly ask NWDAF to obtain current network slice instance information. In both operations, only load level information of slice instances are defined for now. This type of information alone is insufficient to realize a fully automated network with distributed intelligence.

TABLE I: A subset of information at a MEC environment.

Information about
Resources of a Mobile Edge Service
Mobile Edge Subscription resources for a particular subscriber
Mobile Edge Subscription resource for a subscriber
Mobile Edge Traffic Rule resources for an application instance
Mobile Edge Traffic Rule resource
Mobile Edge DNS Rule resources for an application instance
Mobile Edge DNS Rule resource
Mobile Edge Timing Caps resource
Mobile Edge Current Time resource
Available transports

In the standardized 5G architecture (i.e., both reference point-based architecture and service-based architecture), most of the core network information are stored in the unified data repository (UDR). This function is located in the unified data management (UDM). UDR represents a single point of storage for multiple NFs. It holds data from PCF, access and mobility management function (AMF), session management function (SMF), network slice selection function (NSSF), and UDM. Effectively, UDR represents a single point to acquire the various data (i.e., information elements) of the core network and the ongoing flow sessions.

As depicted in Fig. 2, NIF agents, located at the core network management domain, leverage time-series data generated at NWDAF, UDM, UDR, and MEC orchestrator for predicting the variation in the resource utilization, and then accordingly different actions would be taken. NIF agents predict the change in the user plane by leveraging various control plane signals at the NFs above.

5) Service Providers: Different service providers are offering various services to mobile users. Many of these service providers provide an API for developers to extract data from their respective platforms. Most of the APIs communicate using JSON or XML through URL queries, mostly served through HTTP. NIF agents leverage the provided APIs to extract information from the service providers to ensure QoE related to these service providers. Hereunder, we present some standard APIs provided by two popular service providers, namely Facebook, and Google.

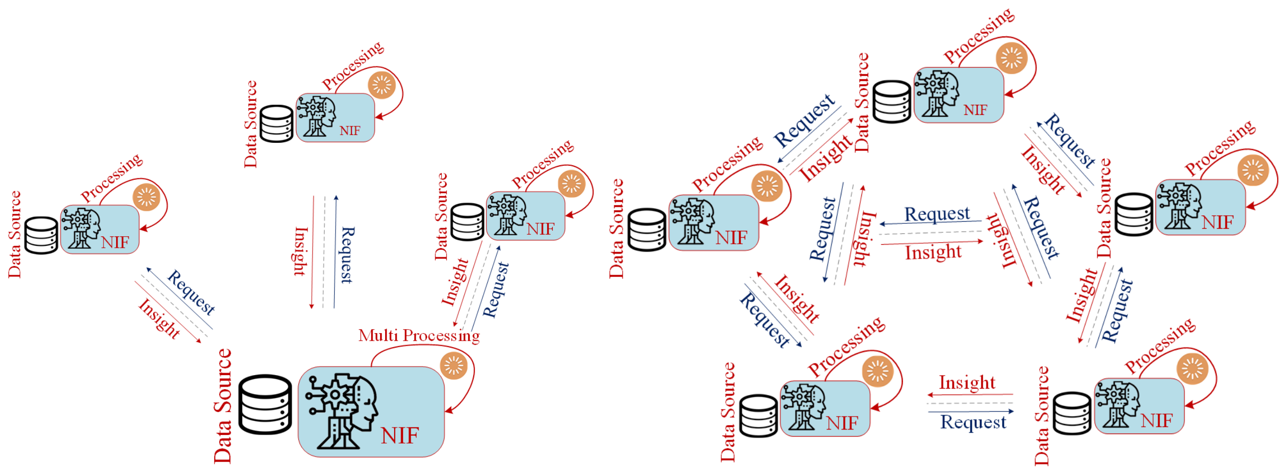
Facebook: An API called ”Graph API”[3] allows a developer retrieve data related to different entities (e.g., Users, Pages, Status, Messages, Videos, and Photos). URLs with a set of filters are used to retrieve the desired data, such as: https://graph.facebook.com/v3.2/⟨node⟩/⟨edge⟩ ?fields=⟨fields⟩&access_token=⟨token⟩

YouTube: Google offers an API for YouTube [4] to retrieve data or manipulate its platform. Since we are more interested

[2]Multi-access Edge Computing (MEC) ETSI Industry Specification Group (ISG), ”Multi-access Edge Computing (MEC); Framework and Reference Architecture”

[3]https://developers.facebook.com/docs/graph-api/overview

[4]https://developers.google.com/youtube/v3/docs/



(a) Centralized Schema

(b) Full Mesh Schema.

Fig. 3: Cross-System AI: Communication Patterns.

in the service providers' data, we focus on the first point. Google offers multiple sets of libraries in different programming languages (e.g., JAVA, JavaScript, PHP, iOS, Go) to access its API. The mean of communication uses a JSON response by accessing REST endpoints with different parameters to filter and obtain the needed data.

C. Domain Integration Fabric, and NIF Agents self-Optimization and self-Collaboration

Fig. 3 depicts the frameworks's communication patterns. Depending on the situation, NIF agents may use different approaches to interact among themselves. In the centralized pattern (Fig. $3(a)$), the NIF agents located at the end-to-end management domain ensure the synchronization among the NIF agents located at the management domains. This approach allows a central unit to orchestrate and manage the different NIF agents. Such global overview facilitates the integration of models and the subsequent decision making. Complex optimization problems can be resolved when the problem can be split into sub-problems, whereby an intelligent agent handles each sub-problem. By selecting suitable agents and combining their knowledge, a solution closer to the optimum can be expected in a relatively short time interval. In this approach, DAI, MAS, FL, MLF, and MARL techniques can be used to synchronize NIF agents.

In the fully-meshed pattern (Fig. $3(b)$), NIF agents communicate directly among themselves. NIF agents, besides their synchronization with the NIF agents located at the end-to-end management domain, can also communicate directly with each other to establish various tasks. NIF agents located in the same management domain can be synchronized as well. In this communication paradigm, NIF agents collaborate to enhance their base of knowledge and accuracy. This pattern is suitable for use cases whereby participating agents are given some autonomy level and are delegated roles of taking local decisions of interest to their respective sub-systems. Autonomous cars and object recognition are two notable examples.

As mentioned in Section II, different AI enablers (i.e., ML, DAI, MAS, FL, MLF, and MARL) have some limitations when considering autonomous self-configuration of complex tasks. For instance, FL and MARL focus more on enabling collaborative decisions for well-specified and straightforward tasks. Moreover, FL distributes the learning process for a simple task with non-i.i.d datasets distributed among NIF agents. The collaborative cross-system AI aims to leverage ML, DAI, MAS, FL, MLF, and MARL techniques to provide a complete framework that can handle complex tasks without strong assumptions, such as the datasets at NIF agents are non-i.i.d. Moreover, different mechanisms can be employed to avoid the global model's bias due to the asynchronization updates of NIF agents.

Learning in parallel and distributed systems requires both plasticity and stability. While plasticity enables integrating new knowledge in parallel and distributed systems, the stability prevents forgetting previously acquired knowledge. The stability-plasticity dilemma is a well-known constraint in DAI. In a collaborative distributed AI system, stability can be achieved by favoring the experienced NIF agents. In contrast, plasticity can be achieved by supporting new NIF agents. Similar to bias-variance trade-off, there is a trade-off between stability and plasticity of the model's underfitting and overfitting. While stability mitigates the problem of underfitting, it can harm overfitting.

Meanwhile, plasticity, similar to the polling layers in deep learning, reduces overfitting risks, but it can harm underfitting. For this reason, in collaborative cross-system AI, an extensive form of strategic game theory is leveraged for enabling a fair trade-off between stability and plasticity. The proposed approach aims to find a fair trade-off between generalization and specialization to overcome the overfitting and underfitting problems. The NIF agents' impact on the collaboration can be autonomously adapted to ensure load balancing between unseen data and the consideration of new agents and new and complex tasks. Moreover, the MLF technique can be leveraged to consider new and complex tasks.

To ensure the collaboration among NIF agents in different management domains, these agents can either share their data sources or combine their knowledge to tackle complex and related tasks. Since each agent has its own AI model, the combination of expertise can be seen as an aggregation of different AI models. To achieve efficient collaboration, several potential methods can be explored:

Majority Voting: Each agent predicts a given instance. This prediction refers to a vote. According to the votes of other agents, the cross-system AI can decide if the agent's vote can be taken into consideration or not.

Stacking ML: This method consists of using another ML algorithm to handle a new dataset generated from the different agents of the system.

Weighting: Depending on the type of information, the source agent and other characteristics, a weight is assigned to each insight received from each participating agent. Weights can be static or dynamically determined. They are decided based on how much influence the system wants to attribute to each agent.

Bootstrap Aggregating: This method consists of converting weak models to strong ones by aggregating the models' multiple predictions to get the final prediction. This technique is mostly used on decision trees in an environment where all agents treat the same data.

IV. Use-case studies

As mentioned previously, NIFs will be integrated into different management domains for enabling smart self-orchestration and self-management of end-to-end network slices that target various emerging use-cases. Service granularity, availability, flexibility, and smartness provided by NIF can benefit a wide range of use-cases that cannot all be described here. We present some use cases and applications that could leverage NIF for enhancing their functionalities and SLA satisfactions.

A. Dynamic Network Customization

In this scenario, the NIF agents are integrated into different components of the network. Those NIF agents, depending on their type, give different kinds of insights, which are gathered by the NIF agents at the end-to-end management domain. These agents have the necessary control on the network to adapt it based on the received information.

Smart Slice Selection and Management: In this context, slice instantiation and attribution to the users is performed based on information collected from the core network, MEC infrastructure, service provider, and user equipment [15]. The idea is to use the NIF agents to collect, analyze and either generate a slice blueprint (in case of slice instantiation) or management related information (e.g., add/remove slice VNF, change slice users). This information gathered by NIF agents is then sent to NSSF, which acts accordingly.

Smart Data Center Energy Optimization: The large energy consumption of the servers in a data center is a significant concern. The servers can be continuously kept running even if no process is running, and that is to be ready to cope with peak hours. By leveraging the collaborative cross-system AI framework to predict and forecast usage peaks, the servers' state can be changed dynamically following a different set of policies (e.g., saving power, medium performance, full performance). This can be achieved by NIF agents that forward the results to the MEC orchestrator that manages the MEC hosts and sets the servers' state according to the information received from the NIF agents.

Smart Policy Planning: Policy planning includes software updates, data backup, and dynamic traffic steering. The next-generation networks will be virtualized/software-based entities. This transition to softwarization will bring its challenges regarding software maintenance with regular updates and software upgrades to new versions that can be incompatible with the other currently deployed entities. Leveraging NIF to plan, identify, validate, and rollback (in case of malfunction) software updates is a way to address these issues.
Data backup can be crucial for critical services, and mistakes can be disastrous if a wrong decision is made. In case of data backup management, a naive decision will be to back up everything, still considering the network's capabilities. However, this may be a costly and inefficient process. In this vein, there is need to plan an efficient strategy for backing up users' data: by considering when to do the backup operation, what to backup prioritizing among different data, and where to place the backups. This can be attained by leveraging a NIF that can take various constraints into account, use different knowledge from other NIF agents (e.g., about data type, network conditions, cloud resource availability, mobility of users, etc), and correlate among the different information to make a logical and intelligent decision.

Smart MEC Application Orchestration: Intelligent management of MEC applications consists of smart MEC application migration (between the VMs or the MEC Hosts), duplication, pausing, releasing, or sharing application information context to other applications. Management decisions are carried out based on various information. Collaborative NIFs offer the possibility for predicting the variation in the network and industrial verticals. Accordingly, the services migrate proactively, or service context is serialized and moved from a MEC to another.

B. Autonomous Car

Autonomous cars can benefit from the collaborative cross-system AI by increasing their capabilities through other cars' combined knowledge. For example, each vehicle can be adopted with a NIF agent with the same goal: finding the best and shortest road to a destination. Those NIF agents deployed on top of these cars communicate with each other and adapt their results by weighting them accordingly. In this scenario, both communication patterns can be adapted to ensure the synchronization among the NIF agents. The decisions that require global knowledge and information from a third party would be offloaded to a central NIF agent located at the cloud or MEC. In this case, NIF agents, on board cars, keep communicating with the central NIF agent for taking the right decisions. From another side, a fully meshed pattern can be also adapted to ensure fast synchronization among the neighboring NIF agents. For instance, the collaboration among the NIF agents enables efficient object detection and tracking and would ultimately yield more accurate actions.

C. Object Recognition

Object recognition problems are resource-consuming and require high computational power. The problem lies in a large number of different 2D/3D images corresponding with a moving object when the lighting changes, or the visibility area changes. To increase the accuracy of the recognition, the collaborative cross-system AI approach could be used. Each NIF agent has a list of recognized objects with their respective accuracy and uses the surrounding agents to increase the different objects' accuracy by leveraging various outputs of NIF agents. In this use-case, both communication patterns can be adapted. A centralized approach can be adopted by offloading the heavy tasks that require more computational to the central NIF. The remaining tasks could be shared among the NIF agents using a fully meshed pattern.

D. Smart Homes

With the significant number of IoT devices present in smart homes, such a scenario aligns in a distributed intelligence paradigm. It could be seen as multiple NIF agents cooperating to achieve a common goal. For instance, a device responsible for regulating the indoor temperature of the house can not only rely on outside temperature but also on the knowledge gained from various other devices (e.g., the device that keeps track of the vital health of the tenants and the monitoring device that keeps records of the past events). The NIF agents, deployed onboard the IoT devices, collaborate for enhancing the sensing and measurement accuracy.

V. Conclusion

This paper leveraged the new achievements in network automation and artificial intelligence for providing a complete framework that enables the collaboration between diverse management domains for enabling end-to-end network slicing. Each network slice is self-orchestrated, self-managed, and self-optimized to meet its desired objectives and KPIs. We have designed our framework to be aligned with the vision of ongoing network standardization. The proposed framework is designed to ensure a cross AI system collaboration for making decisions about multiple tasks in complex scenarios or unseen data. To ensure a fair trade-off between the generalization and the specialization, the proposed framework suggested different techniques for securing a trade-off between stability and plasticity.

Acknowledgment

The research leading to these results received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement n°101016509 (project CHARITY). The paper reflects only the authors' views. The Commission is not responsible for any use that may be made of the information it contains. The research work is also partially funded by the Academy of Finland 6Genesis project under Grant No. 318927, and by the Academy of Finland CSN project under Grant No. 311654. Prof. Song was supported by the Institute of Information communications Technology Planning Evaluation (IITP) (No. 2020-0-00959). Dr. Miloud Bagaa work was supported by CSN project.

References

[1] M. Bagaa, T. Taleb, J. B. Bernabe, and A. Skarmeta, "A machine learning security framework for iot systems," *IEEE Access*, vol. 8, pp. 114 066–114 077, 2020.

[2] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin, K.-W. Wen, K. Kim, R. Arora, A. Odgers, L. M. Contreras, and S. Scarpina, "MEC in 5G networks," European Telecommunications Standards Institute, 06921 Sophia Antipolis CEDEX, France, Whitepaper, June 2018.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *ArXiv*, vol. abs/1312.5602, 2013.

[4] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[5] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 4427–4437.

[6] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "On the convergence of federated optimization in heterogeneous networks," *CoRR*, vol. abs/1812.06127, 2018.

[7] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," 2020.

[8] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 1126–1135.

[9] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *ArXiv*, vol. abs/1911.10635, 2019.

[10] X. Cheng, Y. Wu, G. Min, A. Zomaya, and X. Fang, "Safeguard network slicing in 5g: A learning augmented optimization approach," *IEEE Journal on Selected Areas in Communications*, 2020.

[11] Q. Mahmoud, *Cognitive Networks: Towards Self-Aware Networks*. John Wiley & Sons Ltd, July 2007.

[12] L. Wang and D. T. Delaney, "Qoe oriented cognitive network based on machine learning and sdn," in *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN)*, 2019, pp. 678–681.

[13] V. Sciancalepore, F. Z. Yousaf, and X. Costa-Pérez, "z-torch: An automated NFV orchestration and monitoring solution," *IEEE Trans. Network and Service Management*, vol. 15, no. 4, pp. 1292–1306, 2018.

[14] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Deepcog: Cognitive network management in sliced 5g networks with deep learning," in *IEEE INFOCOM*, 2019, pp. 280–288.

[15] I. Afolabi, J. Prados-Garzon, M. Bagaa, T. Taleb, and P. Ameigeiras, "Dynamic resource provisioning of a scalable E2E network slicing orchestration system," *IEEE Trans. Mob. Comput.*, vol. 19, no. 11, pp. 2594–2608, 2020.

Miloud Bagaa received the engineer's, master's, and PhD degrees from the University of Science and Technology Houari Boumediene, Algiers, Algeria, in 2005, 2008, and 2014, respectively. From 2009 to 2015, he was a researcher with the Research Center on Scientific and Technical Information, Algiers. From 2015 to 2016, he was a Postdoctoral researcher with the Norwegian University of Science and Technology, Trondheim, Norway. From 2016 to 2019, he was a Postdoctoral researcher at Aalto University. From 2019 to 2020, he was a senior researcher at Aalto University. Currently, he is a visiting researcher at Aalto University and a senior system cloud specialist at CSC.

Tarik Taleb [S'05, M'05, SM'10] received the B.E. degree (with distinction) in information engineering in 2001, and the M.Sc. and Ph.D. degrees in information sciences from Tohoku University, Sendai, Japan, in 2003, and 2005, respectively. He is currently a Professor with the School of Electrical Engineering, Aalto University, Espoo, Finland. He is the founder and the director of the MOSA!C Lab (www.mosaic-lab.org).

Jukka Riekki (Dr.Sc.Tech.) is a full Professor of Software Architectures for Embedded Systems at the University of Oulu. He is the Dean of the Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland, and one of the PIs in the Center for Ubiquitous Computing. He holds a Dr.Sc.Tech. in Technology from the University of Oulu and is the author of over 200 peer-reviewed articles. His main research interests are in edge computing, edge AI and intelligent systems. He is actively involved in the 6G flagship program, in developing Fab Lab Oulu, and in developing and applying a new agile research-to-business model in the Allied ICT Finland program.

JaeSeung Song [M'09, SM'17] (jssong@sejong.ac.kr) received the B.S. and M.S. degrees in computer science from Sogang University, and the Ph.D. degree from the Department of Computing, Imperial College London, U.K., in 2013. He is currently an associate professor, leading the Software Engineering and Security Lab (SESLab), Department of Computer and Information Security, Convergence Engineering for Intelligent Drone, Sejong University.