Network Artificial Intelligence, Fast and Slow

Dario Rossi
Huawei Technologies, co. Ltd
Paris, France
dario.rossi@huawei.com

Liang Zhang
Huawei Technologies, co. Ltd
Nanjing, China
zhangliang1@huawei.com

ABSTRACT

As networks have historically been built around connectivity, architectural features concerning quality of service, mobility, security and privacy have been added as afterthoughts – with consequent well known architectural headaches for their later integration. Despite Artificial Intelligence (AI) is more a means to an end, that an architectural feature itself, this is not completely different from what concerns its integration: in particular, while Cloud and Edge computing paradigms made it possible to use AI techniques to relieve part of network operation, however AI is currently little more than an additional tool. This paper describes a vision of future networks, where AI becomes a first class commodity: its founding principle lays around the concept of "fast and slow" type of AI reasoning, each of which offers different types of AI capabilities to process network data. We next outline how these building blocks naturally maps to different network segments, and discuss emerging AI-to-AI communication patterns as we move to more intelligent networks.

KEYWORDS

AI Native Networking, Machine Learning

1 INTRODUCTION

Over the last 60+ years, networking evolved from a mono-application circuit-switched telecommunication technology, to a multi-application all-IP cloud-native architecture. To fully embrace Artificial Intelligence (AI) benefits, coming e.g., from Machine Learning (ML) and Deep Learning (DL) models, the next step in networking evolution will be to place the latter at the very heart of its architectural definition, giving birth to "AI native" networks.

As illustratef in Fig. 1, AI and network technologies evolved in parallel since the early 60: while early cross-pollination attempts dates back the early 70s[1], they were tepidly received, and the 1st AI winter froze further early-adoption. During the 80s, AI and networking independently evolved, laying out the basis of the current successful ML and IP technologies – once more facing difficulties such as a 2nd AI winter in the late 80s, or the burst of the .com bubble at the wake of the new millennium. Since the early 2000 (resp. 2010), ML (resp. DL) and networking crossed paths again: in these recent research waves, the focus has generally been on how to leverage advances in AI technologies to solve specific networking tasks (that are surveyed e.g., in [2–4]): i.e., with very limited exceptions, AI has been merely used as an addendum to the network architecture, that we can refer to as "AI Assisted" networking. Yet, it now starts being recognized [5] that the network architecture should evolve to embrace a deeper synergy with AI to fully harness its benefit, which would thus become a more fundamental network building block. Otherwise stated, in the upcoming "AI native" network architecture, AI would no longer be a late addendum, but would rather be the starting point of the equation, leading to the confluence of networking and AI in a more inter-wined evolution path.

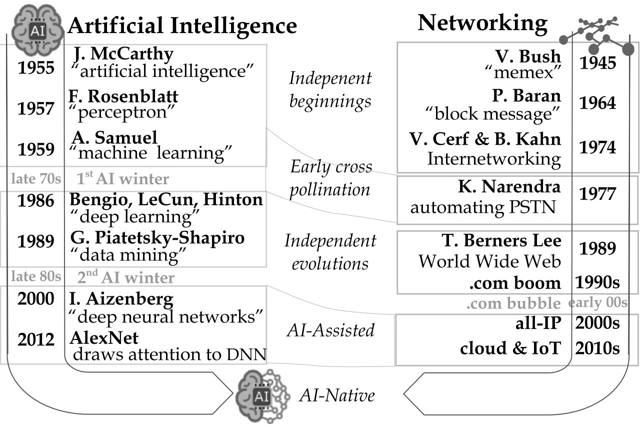

Figure 1: Graphical timeline of the evolution of the Artificial Intelligence and Networking domains.

In this paper, we outline and discuss a vision for native network intelligence. In particular, after overviewing related work (Sec. 2), we refine the so far quite fuzzy definition of "AI Native" networking (Sec. 3). We next instantiate what we believe should be the guiding principles for native network intelligence, in a framework that remains voluntarily at high-level (Sec. 4): by divide and conquer, we make a stark contrast between "fast" and "slow" AI tasks, which are respectively useful to (i) pervasively leverage AI knowledge for enhanced perception and frequent decisions, and (ii) to continuously advance AI knowledge as a slow yet steady background task.

2 BACKGROUND

While much research has been carried out on "AI Assisted" networking in the last decades [2, 3, 6] the "AI Native" networking is still very much in its infancy, leaving the concept open to interpretation from normalization and funding bodies [5, 7–9] to industrial [10–12] and academic contexts [13–18].

AI native in in Standards and Funding bodies. Several standards appeared that are related to AI, which are comprehensively overviewed by the EU Observatory for ICT standardization [7]: despite a fraction is also related to networking technologies, as in particular ETSI Experiential Network Intelligence[8], ETSI Zero-Touch Network Service Management and IRTF Computing in the Network[9], no standard emerged yet that has the ambition of defining an AI native network architecture.

Yet, the need for architectural changes is recognized, for instance in the EU HORIZON program by funding research and innovation actions addressing the native integration of AI for telecommunications, to *"implement adaptive decision making at different time scales with expected [...] changes in the existing architectures."* [5]. In terms of EU funded projects, the closest in scope is the H2020 project on "Network intelligence for adaptive and self-learning mobile networks" (DAEMON)[19], that considers how to integrate AI over different use cases spanning the whole end-to-end architecture.

AI native in Industrial whitepapers. Smarter customer products [10] are among the first references of "AI Native" in the industry, while closer to the networking field, Ericsson [11] and DeepSig [12] both explore AI Native in the context of 5G technologies. In particular, Ericsson targets 5G Business Support Systems, considering e.g., the reduction of manual effort in the invoicing [11]. Instead, DeepSig exploits ML for signal processing [12] in the Open Radio Access Network (O-RAN) Distributed Unit (O-DU) of the 3GPP Rel 18 standard — which is a natural evolution of the Self-Organizing Networks (SON) pushed by the 3GPP and the NGMN organizations.

AI native in Academic research. In light of the previous observations, it is not a surprise that AI Native networking research emerged in a 5/6G context, with special emphasis on the communication aspects [13], air-interface [14], radio access [15] or slicing [16], with a push toward *edge* [17] or *in-network* [18] intelligence. These effort have merit as they systematically expose challenges and breakdown requirements for AI native 5G communication [17], offering some architectural design [15] to facilitate the use of AI for multiple aspects of air transmission – however with few exceptions [18] the lessons are difficult to be transposed outside of the realm of PHY communication for 5G/6G (or WiFi-7, though not explicitly considered in any of the above). We further remark that, while similar effort is undergoing in the broader computer science field (as, e.g., for the case of AI Native databases [20]) however it is manifest that little can be adapted from different fields (as the optimization or redesign of index maintenance, query or join operations with AI are ultimately specific to the database community).

3 PRINCIPLES OF AI NATIVE NETWORKS

3.1 Viewpoints and definitions

To structure the discussion of Native Network Intelligence(NNI), we start by observing that its definition can lend itself to multiple interpretations, i.e., where the network:

① ***deeply integrates AI*:** e.g., when fundamental tasks such as 6G signal processing [13–17], routing [21], packet classification [22] or bloom filters [23] are realized or augmented with learning techniques;

② ***systematically leverages* AI:** e.g., in architectures [15, 19, 24] where AI is a fundamental, thought not the only, part of the decisional tools;

③ **is *designed by* an AI:** e.g., when protocol (operational points) are learned, as in the case of TCP [25–27], instead of being carefully heuristically tuned as in the numerous TCP variants surveyed in [28, 29]);

④ **is *designed for* AIs:** e.g., AI-to-AI communication where endpoints are AI agents developing their own languages [30, 31], as opposite to humans or machines of IoT/industrial settings.

We next observe that the above viewpoints are not mutually exclusive, but are rather complementary, possibly imposing different requirements for an native network intelligence. This also follows from the fact that, generally speaking, any NNI needs a comprehensive and holistic vision: i.e., instead of considering AI/ML as tools to solve specific problems in an isolated manner, the main goal of an NNI architecture is to more organically integrate AI in the system itself, so to more coherently and cohesively exploit AI/ML across the whole spectrum of tasks for which it may be needed.

At the same time, the risk in defining an NNI architecture is to put excessive emphasis on AI as being the unique building block able to solve *everything*, which does not hold. This is well illustrated in Fig.2, which clearly scopes the role of AI: albeit important, AI is just a *tiny piece* of the whole Google infrastructure [32] – and the fact that this holds for an hyper-giant that makes intensive use of Cloud AI, strongly suggests this is therefore likely to hold for the *networking* field too. Otherwise stated, an NNI architecture should start from the tight integration of *all pieces*, and not myopically focusing on just some (i.e., AI) but forgetting equally important others (e.g., data representation, resources and infrastructure).

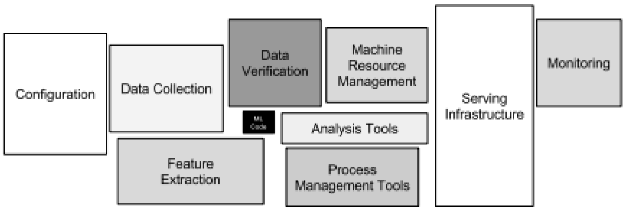

Figure 2: The role of AI in the overall Google architecture as overtly discussed in the AI community: *"Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex"* [32]. Picture courtesy of [32].

3.2 Goals and implications

We next list a few[1] foundational goals for an NNI, that are related to the above viewpoints.

3.2.1 Robust AI: Systematic lifelong and safe learning. Networks currently lack universal performance models, owing to complexity of producing such models with analytical techniques [33], for which the old adage is that "all models are wrong, but some are useful" (G. Box). As even simulation may face fidelity problems, data-driven models appear to be an appealing alternative: at the same time, AI models are intrinsically tied to the data they have been trained with, so that they can be possibly significantly wrong in some deployments. Otherwise stated, fairness issues and bias of AI models in the network field means they can have bad generalization properties in practice, so they will need to be constantly updated and verified. This is particularly hazardous for models that need to actuate in the network, so that safe exploration techniques are necessary to learn in a robust fashion. Robust and lifelong learning is thus a precondition to enable a ①- ② systematic use of AI in networks.

3.2.2 Trustworthy AI: Explainable and accountable. It is now quite well recognized that a critical step for AI to be perceived as safe, and therefore be accepted as trustable, is to be understood: this is necessary to avoid "Clever Hans" models that, e.g., are right but for the wrong reasons [34]. The notion of trust can get a different connotation depending on the target audience: researchers may be questioning model fairness and data imbalance, operators may be interested in understanding model limits in their deployment, some form of accountability may be needed for business viewpoint, while the ability to provide accurate step-by-step explanation may be required for legal compliance due to regulator bodies [35]. Compliance verification need to be done without leaking sensitive model aspects, but may require faithful explainations [36], i.e., an accurate account of the actual decisions taken by a model, as opposite to as approximate explanation of surrogate models [37]. As for other fields [38], trust is especially critical for growing AI usage for network ③ design and operation ①-②.

3.2.3 Configuration of AI: Simplicity first. Additionally, network configuration is knowingly already quite complex. While AI can be used to possibly automate part of the configuration[39], as AI models itself need to evolve, there is an expressiveness tradeoff between hiding most of AI complexity (with the downside of blackbox obscurity), vs explicitly exposing inner configuration AI parameters as as further parameters (exposing thousands, if not millions DL weights, which is clearly not desirable either). To simplify AI deployment, some go to the point of proposing to significantly hide AI in production by using *declarative* interfaces (as in the Ludwig and Overton systems [40]) which is not without appeal, as it would simplify the use of existing models while still allowing for sophisticated control for lifelong learning – which seems to be desirable to break an adoption barrier, and retain the necessary level of flexibility to ① deeply integrate and ③ retain fine-grained control over AI at the same time.

3.2.4 Open vs proprietary AI: Business coexistence. Irrespectively of how AI models are going to be configured, trained and deployed in practice, it is clear that having good quality models is of primary importance. While vendors are obvious candidates for providing proprietary "defaults" models, the general tendency in the AI field is to offer API for plugging third-party models, and optionally ③ combine such "learnware"[41], which is tempting yet more challenging. For the networking field, is too early to understand the business viability of model marketplaces [42] and open formats [43], yet it seems reasonable to at least allow for a coexistence of both approaches (i.e., custom silos vs models co-developed within a partner ecosystem), as this can lead to an increase of a wider spectrum (and possibly higher quality) of learnware models.

3.2.5 Flexible architecture: Heterogeneous topological and incremental deployment. AI deployment in a network is topologically more articulated than the current model of device AI (e.g., smartphone app) plus Cloud AI (i.e., the app's backend server): as such, NNI faces additional complexity concerning where AI functions need to be instantiated. For instance, Cloud AI may increase model generalization capabilities for the vast majority of (but not all) networks, so that local specialization would be preferable; at the same time, network equipment is generally less powerful that the current breed of user devices, so that while access/CPE AI may impose

[1] Due to space constraint, we do not cover equally important aspects tied, e.g., to automation, hardware support, AI execution and certification.

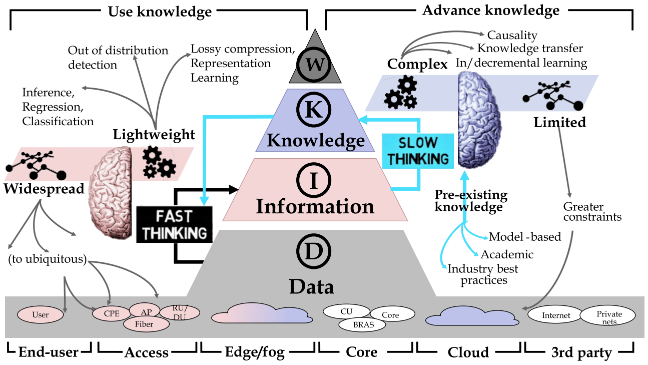

Figure 3: Network DIKW Pyramid vs "Fast and slow" thinking: "fast" AI is pervasive and exploits current knowledge, while "slow" AI is more sparingly used to incrementally advance knowledge.

severe constraints to learning/inference, Edge/Fog AI may represent a good compromise between these two extreme.

As constraints may heavily depend on use-case, it follows that to support ① at architectural level, the NNI should be flexible in mapping a large heterogenity of AI functions to possibly different network segments – though even recent advances in serverless [44] execution models [45] are not sufficient to fully address NNI deployment concerns. Strictly related, it is important for the NNI architecture to allow for incremental deployment – to avoid, e.g., a chicken-and-egg problem concerning the need for new equipment, resource or capabilities to be deployed at very specific points/segments in the network prior that any benefit can be obtained, which could otherwise compromise adoption, hampering ②.

3.2.6 Flexible primitives: AI-2-AI communications. Unsurprisingly, NNI systematically exploits AI to optimize network operation, chaining a set of AI function executed at different nodes, that we can refer to as AIFV. Additionally, these AIFV can be seen as a set of communicating AI agents, irrespectively of whether their purpose is related to network operation: thus, the type and role of these ④ inter-AI communication should not be constrained a priori – as TCP/IP did not constrain applications. In turn, this may push to break the boundaries of classic "effective yet rigid" communication protocols, by introducing new primitives tailored to AI agents need. For instance, while two software endpoints communicate with in-order lossless compression, two AI learnwares may favor lossy data representation to coordinate distributed decision (e.g., sampling over time[46] or space, via feature selection or non-linear embedding). Thus, the AIFV paradigm should offer in-network computing primitives for AI-to-AI communication (e.g., with support for lossy compression, which can be itself a learned block, as opposite to a manual design [46] fit for a single purpose).

4 NNI BUILDING BLOCKS

4.1 High-level concepts

4.1.1 From data to knowledge. We start introducing high-level concepts that are useful for NNI architecture definition. Generally speaking, AI techniques are useful to (i) enhance *perception* by organizing data into information, and (ii) take *decisions* according to knowledge. Figure 3 illustrates the classic Data, Information, Knowledge and Wisdom (DIKW) Pyramid, and additionally highlights the two "Fast" and "slow" type of AI-reasoning discussed next.

As for (i) enhanced perception, differently from other fields (e.g., self driving cars) where AI is heavily used to make sense of physical world sensors (e.g., from lidar and camera readings to moving cars on a map), fine-grained network telemetry is abundant: hence, perception tasks are essentially distilling information out of large volumes/streams of data (e.g., clustering, anomaly detection, important sampling).

In terms of (ii) knowledgeable decisions, the need for a Knowledge plane has been long recognized [47], yet it only rarely gained implementation maturity (end even when it did, implementations were mostly limited to measurement systems [48, 49] with very limited and specific intelligence[50]). In light of recent advances in AI techniques, it is thus useful to reassess the concept of network "knowledge". While the comparison is voluntarily stretched, Jean Piaget famously said that "Intelligence is not what you know, but what you do when you don't know": in the NNI, AI should be aware of the limit of its knowledge, and strive to continuously improve it.

According to Socrates, knowledge advances with a 2-stages maieutic method, which in the first phase raises doubts about what is known, and the second phase advances the knowledge itself. In this position paper, we posit that the above method should also apply to NNI. These two phases should be reflected in the lifelong learning process of an NNI with the first phase recognizing that the models in use have some limits (i.e. detecting out-of-distribution samples, such as zero-day traffic/attacks, or patterns that are very different than those seen at training times), and either exposing these problems to human operators (to gain trust, which needs explanation capabilities), or automatically closing the loop by generating new knowledge (e.g, after trust is fully gained, by updating models to incorporate for concept drift of old classes, or entirely new classes, new facts, etc.).

4.1.2 System-1 vs System-2. We now introduce a knowingly inaccurate[2] analogy, about the need for "Fast and slow"[52] thinking skills, to realize the above DIKW-related tasks in the NNI. According to psychology research popularized by [52], the human brain is organized into two subsystem. System-1

[2] As the psychobehavioral model in [51] was developed to explain irrational bias in human decision related to economics.

is fast and correct on simple repetitive tasks: using it has low cost, but it may be prone to bias and errors. In case of human beings, psychological bias (such as risk adversion, illusory correlation, insensitivity to previous outcomes, etc.) can affect their behavior in non-rational ways, but this is without serious consequences for most of their daily decisions. System-2 possesses advanced capabilities required for complex tasks, or to take decision in situations with missing information: using System-2 is slow, requires a significant cognitive effort, and is thus used more parsimoniously.

Extending this analogy to the case of NNI, we observe that many actions require fast timescales and need prompt responses. AI bias for network models (e.g., due to lack of contexual or broader information, ageing models or concept drift) can affect decisions in sub-optimal ways, yet the approximation may be not noticeable (or can be tolerable) in most of the cases. In the NNI, "fast" intelligence should therefore be pervasive, as we can expect the benefits of using (perhaps slightly inaccurate) models to largely out-weight the loss of not using them. System-1 is thus necessary for a deep, systematic and native integration of AI in networks.

At the same time, is imperative that the "fast" models in use are not stale (e.g., in changing environments) or just plainly unfit (e.g., for a new protocol, application, environment, spectrum, colored noise), which confirms the need for System-2 capabilities in the NNI. Given that these advanced knowledge generation capabilities are used at a slower rate and may benefit from a global view [47], a widespread deployment does not seem to be necessary (i.e., Cloud/Edge/Fog should suffice for most of the use-cases). At the same time some System-2 capabilities (e.g., for fine tuning relatively simple models with local-only significance) may benefit from in-device availability (e.g., saving the bandwidth/time cost of moving toward the Fog/Edge/Cloud). System-2 is thus necessary to support AI lifelong and robust learning, and to additionally make AI a trusted tool for the network domain (e.g, recognize and avoid bias, explain decisions with causality arguments and avoiding spurious correlation).

4.1.3 AI-to-AI communication. We finally abstract the set of AI functions that are chained and executed in the network, and instead of focusing on few relevant yet specific use-cases, we analyze AIFV under the light of a more general AI-to-AI communication paradigm, by describing the set of emergent patterns that we expected to be frequent in the NNI, which are portrayed in Figure 4.

We highlight that NNI fundamentally differs from the current AI-to-AI communications in Cloud AI settings: these are well exemplified by the "federated learning" paradigm, introduced by Google [53] and that recently enjoyed a growing popularity, illustrated in Figure 4(e). In federated learning, relatively powerful devices (e.g., smartphones) leverage Cloud resources for model fusion, yet the communication remains OTT and all network devices in the path are oblivious to the process: as the exchanged model weights are typically private between the two System-2 endpoints, the whole network remains a dumb pipe during the whole lengthy process.

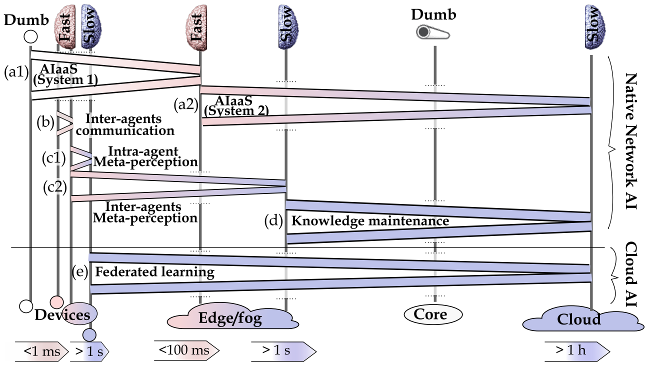

Figure 4: AI-to-AI communication pattern: Fast and Slow thinking among different NNI segment/endpoints vs current Cloud-native AI.

In contrast, NNI AI-to-AI patterns reported in Figure 4(a)-(e) are richer, owing to the network topological complexity. While in spirit with original TCP/IP architecture we believe that the network core should remain stateless and simple[54] (i.e., we are not advocating the NNI as a resurrection of active networks[55]), we envision AI-to-AI communication between (i) multiples set of end-points in the network, as well as (ii) multiple type of intelligence. For instance, dumb terminals may require AI as a service, e.g., for System-1 perception or decision tasks (a1); even a relatively powerful Edge/Fog compute with regional visibility of a fleet of devices, may require System-2 capability in the faraway Cloud for broader visibility (a2). For sufficiently simple setups, fast and slow intelligence can be available in devices (b,c1). For instance, inter-agent communication (b) can happen e.g., for distributed decision making among System-1 agents in embedded systems. Communication between System-1 and System-2 (indicated as "meta-perception") is expect to happen regularly, e.g., for local tuning in the intra-agent case (c1), or to inform the Fog/Edge and invoke its assistance for knowledge update (c2). The closest communication pattern to Cloud-native federated learning is denoted as "knowledge management" (d), where e.g., Fog/Edge System-2 involve the Cloud for cross-network generalization of global concepts.

4.2 Low-level tools

4.2.1 AI knowledge management (KM). Ultimately, high-level concepts need to materialize as low-level implementation, that we start discussing. Differently from other application domains where the goal is to mimic (or beat) human thinking (on writing, drawing, gaming skills), intelligent networks do not need the emergence of general AI. Yet, the very same definition of knowledge in the NNI, and the selection of the tools apt for its management is far from being straightforward, which we separate into Monolithic vs Holistic KM.

Monolithic KM is the maintenance of specific ML/DL models for narrow objectives. This can already be complex as the model catalog can be organized by a single entity with tight grip on every aspect (the "cathedral") or by a larger ecosystem of contributing entities (the "bazaar" or, in AI domain, model marketplaces [42]). For this necessary (but not sufficient) task, tools with open API exist (e.g., ONNX [43], CoreML [56]), which makes monolithic KM possible for vendors, 3d-parties and tech-savvy customers.

Holistic KM has a broader viewpoint, and addresses interdependence across the whole spectrum of models for orchestration, which requires a set of tools (expert models, knowledge graphs, ontologies) that have enjoyed a more timid succeed compared to other domains. Holistic KM faces additional difficulties (e.g., open marketplaces, multi-vendor interoperability, etc.), bringing problems related to knowledge transfer. We further stress that "holistic" does not rhyme with "centralized": i.e., NNI cannot have a single "brain" equipped with slow thinking capabilities, yet a heavily distributed [47] architecture is currently far from being desirable. It hence follows that multiple hierarchically connected peripheral brains may provide a good starting point, with their interconnection topology [57] trading off latency vs communication costs for holistic KM.

4.2.2 AI native representation of network data. Knowledge needs data to reason about, which abunds in NNI. As, from an information viewpoint there are diminishing returns in accessing to *all* data, learning a NNI representation can tradeoff telemetry bandwidth, information content and processing cost. At the same time, network data is multi-modal, multivariate in each mode, temporally multi-scale and topologically multi-layer – which makes finding a natural representation far from being trivial. For the NNI, the question is how to transform *named entities* (e.g., devices, IP addresses, TCP flows, named services, etc.) and corresponding *categorical and numerical quantities* characterizing them (e.g., configuration, throughput, KPIs, etc.) into an AI native representation fitter for AI processing and decision making: this common representation would facilitate (fast) decision making and (slow) knowledge improvement. While such unified NNI data representation is lacking, it is plausible to assume that it would be composed of heterogenity of input types, each of which requires specific AI processing (e.g., from random forests for tabular data, to recurrent networks for timeseries, to transformers for sequences, to graph neural networks for topologies), which is not yet clear how to best combine [58].

4.2.3 AI execution and triggers. Key to deployment are AI execution model, and the intimately related aspect of how AI functions are best triggered from/within the network.

Execution Many alternatives exist for executing AI functions, including container, severless [44] and Function as a Service (FaaS) models, which fits Cloud/Edge/Fog segments. While research is ongoing to accelerating stateless [45] or stateful [59] FaaS, or accelerating ML stacks (such as TensorFlow) for constrained devices (such as TL-lite, TFlite-micro[60]), these still leverage (relatively) powerful ARM CPUs or domain specific hardware accelerators. The question on what can be done on more constrained embedded network devices (e.g., low power ARM cores, P4 match-action tables) and which execution model is the fittest (e.g., bytecode, micro-kernel, micro-code) is still open. To address this, we argue that NNI will have to impose a more Network-centric execution model, that still needs to integrate with KM (though not necessarily with marketplaces) for securely deploying updated models to all network elements.

Triggers Related question concerns calls to AI functions: if RESTful APIs are again fit for Edge/Cloud functions, however they would result in a clear overkill for in-network operation. The AIFV paradigm introduced early can be wired to, e.g., a IP-native implementation, where the AI-to-AI processing chain exploits SRv6 routing capabilities to trigger AI functions, extending the current SRv6 programmability model [61]. At the same time, we ought to stress that while SRv6 triggers may bring *parameters* and *state* (e.g., piggybacked in header extension, to circumvent stateless function limits[59]) we argue that they should not carry *code* or *instructions* – i.e., the active network [55] paradigm and its recent resurgence [62] may be a good fit for network programming, but is in our opinion far from NNI needs.

5 CONCLUSIONS

This position paper sketches a high-level architectural picture of Native Network Intelligence (NNI), leaving numerous implementation details, where we are certain that hordes of devils hide, for future work.

Our main idea is to simplify architectural decision with a *divide et impera* argument, whose starting point is to recognize that (i) advanced perception and decisions tasks are frequent enough, simple enough and good enough to justify pervasive in-network deployment; conversely, tasks related to (ii) knowledge management are critically important, yet slower and less frequent, possibly requiring the assistance of Fog/Edge/Cloud. We further recognize that (i) and (ii) are two sides (endpoint) of the same coin (communication), and discuss the rich(er) set of patterns of AI-to-AI communication that we expect to emerge in the NNI (vs Cloud AI), opening new opportunities as well as new research challenges.

REFERENCES

[1] K. S. Narendra et al. "Application of learning automata to telephone traffic routing and control". In: *IEEE Trans. SMC* 7.11 (1977).

[2] R. Boutaba et al. "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities". In: *JISA* 9.1 (2018).

[3] H. Lutfiyya et al., eds. *IEEE TSNM SI on Embracing AI for Network and Service Management.* Vol. 18. 2021.

[4] D. Rossi and L. Zhang. "Landing AI on Networks: An equipment vendor viewpoint on Autonomous Driving Networks". In: *IEEE TNSM* 19.3 (2022).

[5] https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/horizon-ju-sns-2022-stream-b-01-01.

[6] P. Chemouil et al., eds. *IEEE JSAC SI on Advances in AI and ML for Networking*. Vol. 38. 2020.

[7] L. Frost et al., eds. *Report of TWG AI: Landscape of AI Standards.* 2021. DOI: 10.5281/zenodo.4775836.

[8] https://www.etsi.org/technologies/experiential-networked-intelligence.

[9] https://datatracker.ietf.org/rg/coinrg/about/.

[10] https://medium.com/@larsbuttler/power-to-the-people-a-i-native-ad634a65871c#.eb23bp3cg. 2016.

[11] https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/bss-and-artificial-intelligence-time-to-go-native. 2019.

[12] https://www.deepsig.ai/ai-native-communications. 2022.

[13] G. Wu. "6G Technologies for Mobile Connected Intelligence". In: *ITU Kaleidoscope (ITU-K).* 2021.

[14] J. Hoydis et al. "Toward a 6G AI-native air interface". In: *IEEE Communications Magazine* 59.5 (2021), pp. 76–81.

[15] P. Carbone et al. "NeuroRAN: rethinking virtualization for AI-native radio access networks in 6G". In: *CoRR* abs/2104.08111 (2021).

[16] W. Wu et al. "AI-native network slicing for 6G networks". In: *IEEE Wireless Communications* 29.1 (2022), pp. 96–103.

[17] Y. Xiao et al. "Toward self-learning edge intelligence in 6G". In: *IEEE Communications Magazine* 58.12 (2020), pp. 34–40.

[18] X. Li et al. "Advancing Software-Defined Service-Centric Networking Toward In-Network Intelligence". In: *IEEE Network* 35.5 (2021).

[19] https://h2020daemon.eu/. 2022.

[20] G. Li et al. "XuanYuan: An AI-Native Database." In: *IEEE Data Eng. Bull.* 42.2 (2019), pp. 70–81.

[21] E. Gelenbe et al. "Self-aware networks and QoS". In: *Proceedings of the IEEE* 92.9 (2004).

[22] E. Liang et al. "Neural packet classification". In: *ACM SIGCOMM.* 2019.

[23] K. Vaidya et al. "Partitioned Learned Bloom Filters". In: *ICLR.* 2021.

[24] D. Harel et al. "Autonomics: In search of a foundation for next-generation autonomous systems". In: *PNAS* 117.30 (2020).

[25] K. Winstein and H. Balakrishnan. "Tcp ex machina: Computer-generated congestion control". In: *ACM SIGCOMM.* 2013.

[26] X. Nie et al. "Dynamic TCP initial windows and congestion control schemes through reinforcement learning". In: *IEEE JSAC* 37.6 (2019).

[27] S. Emara et al. "Eagle: Refining congestion control by learning from the experts". In: *IEEE INFOCOM.* 2020.

[28] J. Widmer et al. "A survey on TCP-friendly congestion control". In: *IEEE network* 15.3 (2001).

[29] R. Al-Saadi et al. "A survey of delay-based and hybrid TCP congestion control algorithms". In: *IEEE Comm. S&T* 21.4 (2019).

[30] A. Lazaridou et al. "Multi-Agent Cooperation and the Emergence of (Natural) Language". In: *CoRR* abs/1612.07182 (2016).

[31] I. Mordatch and P. Abbeel. "Emergence of Grounded Compositional Language in Multi-Agent Populations". In: *AAAI* (2018).

[32] D. Sculley et al. "Hidden Technical Debt in Machine Learning Systems". In: *NeurIPS.* 2015.

[33] M. Ferriol-Galmés et al. "Routenet-erlang: A graph neural network for network performance evaluation". In: *IEEE INFOCOM.* 2022.

[34] S. Lapuschkin et al. "Unmasking Clever Hans predictors and assessing what machines really learn". In: *Nature* 10.1 (2019).

[35] https://www.europarl.europa.eu/thinktank/en/document/EPRS_BRI(2021)698792.

[36] P. J. Phillips et al. "Four principles of explainable artificial intelligence". In: *NIST Draft* (2020).

[37] Z. Meng et al. "Interpreting Deep Learning-Based Networking Systems". In: *ACM SIGCOMM.* 2020.

[38] R. R. Hoffman et al. "Trust in automation". In: *IEEE Intelligent Systems* 28.1 (2013).

[39] Z. Ben Houidi and D. Rossi. "Neural language models for network configuration: Opportunities and reality check". In: *Computer Communications* 193 (2022).

[40] P. Molino and C. Ré. "Declarative machine learning systems". In: *Commun. of the ACM* 65.1 (2021), pp. 42–49.

[41] Z.-H. Zhou. "Learnware: on the future of machine learning." In: *Frontiers Comput. Sci.* 10.4 (2016), pp. 589–590.

[42] https://www.acumos.org/.

[43] https://onnx.ai/.

[44] E. Jonas et al. "Cloud Programming Simplified: A Berkeley View on Serverless Computing". In: *CoRR* abs/1902.03383 (2019).

[45] S. Kotni et al. "Faastlane: Accelerating Function-as-a-Service Workflows". In: *USENIX ATC.* 2021.

[46] N. Yaseen et al. "Towards a Cost vs. Quality Sweet Spot for Monitoring Networks". In: *ACM HotNets.* 2021.

[47] D. D. Clark et al. "A knowledge plane for the internet". In: *ACM SIGCOMM.* 2003.

[48] H. V. Madhyastha et al. "iPlane: An information plane for distributed services". In: *USENIX OSDI.* 2006.

[49] A. Shieh et al. "NetQuery: A knowledge plane for reasoning about network properties". In: *ACM SIGCOMM.* 2011.

[50] A. Mestres et al. "Knowledge-defined networking". In: *ACM SIGCOMM Computer Communication Review* 47.3 (2017), pp. 2–10.

[51] A. Tversky and D. Kahneman. "Judgment under Uncertainty: Heuristics and Biases". In: *Science* 185.4157 (1974).

[52] D. Kahneman. *Thinking, fast and slow.* 2011.

[53] B. McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: *PMLR AI and statistics.* 2017.

[54] V. Cerf and R. Kahn. "A protocol for packet network intercommunication". In: *IEEE Trans. Comm.* 22.5 (1974).

[55] D. L. Tennenhouse and D. J. Wetherall. "Towards an active network architecture". In: *ACM SIGCOMM* (1996).

[56] https://github.com/apple/coremltools.

[57] O. Marfoq et al. "Throughput-optimal topology design for cross-silo federated learning". In: *NIPS.* 2020.

[58] Z. B. Houidi et al. "Towards a systematic multi-modal representation learning for network data". In: *ACM HotNets.* 2022.

[59] V. Sreekanti et al. "Cloudburst: Stateful Functions-as-a-Service". In: *VLDB* (2020).

[60] R. David et al. "TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems". In: *MLSys* (2021).

[61] C. Filsfils et al. "SRv6 network programming". In: *IETF RFC 8986* (2021).

[62] J. Xing et al. "A Vision for Runtime Programmable Networks". In: *ACM HotNets.* 2021.