Workgroup: Internet Research Task Force
Internet-Draft:
draft-francois-nmrg-ai-challenges-02
Published: 13 March 2023
Intended Status: Informational
Expires: 14 September 2023
Authors: J. François
University of Luxembourg and Inria
A. Clemm
Futurewei Technologies, Inc.
D. Papadimitriou
3NLab Belgium Reseach Center
S. Fernandes
Central Bank of Canada
S. Schneider
Digital Railway (DSD) at Deutsche Bahn

Research Challenges in Coupling Artificial Intelligence and Network Management

Abstract

This document is intended to introduce the challenges to overcome when network management problems may require to couple with AI solutions. On the one hand, there are many difficult problems in Network Management that to this date have no good solutions, or where any solutions come with significant limitations and constraints. Artificial Intelligence may help produce novel solutions to those problems. On the other hand, for several reasons (computational costs of AI solutions, privacy of data), distribution of AI tasks became primordial. It is thus also expected that network **SHOULD** be operated efficiently to support those tasks.

To identify the right set of challenges, the document defines a method based on the evolution and nature of NM problems. This will be done in parallel with advances and the nature of existing solutions in AI in order to highlight where AI and NM have been already coupled together or could benefit from a higher integration. So, the method aims at evaluating the gap between NM problems and AI solutions. Challenges are derived accordingly, assuming solving these challenges will help to reduce the gap between NM and AI.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 14 September 2023.

Copyright Notice

Table of Contents

1. Introduction

The functional scope of network management (NM) is very large, ranging from monitoring to accounting, from network provisioning to service diagnostics, from usage accounting to security. The taxonomy defined in [Hoo18] extends the traditional Fault, Configuration, Accounting, Performance, Security (FCAPS) domains by considering additional functional areas but above all by promoting additional views. For instance, network management approaches can be classified according to the technologies, methods or paradigms they will rely on. Methods include common approaches as for example mathematical optimization or queuing theory but also techniques which have been widely applied in last decades like game theory, data analysis, data mining and machine learning. In management paradigms, autonomic and cognitive management are listed. As highlighted by this taxonomy, the definition of automated and more intelligent techniques have been promoted to support efficient network management operations. Research in NM and more generally in networking has been very active in the area of applied ML [Bou18].

However, for maintaining network operational in pre-defined safety bounds, NM still heavily relies on established procedures. Even after several cycles of adding automation, those procedures are still mostly fixed and set offline in the sense that the exact control loop and all possible scenarios are defined in advance. They are so mostly deterministic by nature or or at least with sufficient safety margin. Obviously, there have been a lot of propositions to make network smarter or intelligent with the use of ML but without large adoption for running real networks because it changes the paradigms towards stochastic methods.

ML includes regression analysis, statistical learning (SVM and variants), deep learning (ANN and variants), reinforcement learning, etc. It is a sub-area of AI that concentrates the focus nowadays but AI encompasses other areas including knowledge representation, inductive logic programming, inference rule engine or by extension the techniques that allow to observe and perform actions on a system.

It is thus legitimate to question if ML or AI in general could be helpful for NM in regards to practical deployment. This question is

actually tight with the problems the NM aims to address. Independently of NM, ML-based solutions were introduced to solve one type of problems in an approximate way which are very complex in nature, i.e. finding an optimal solution is not possible (in polynomial time). This is the case for NP-hard problems. In those cases, solutions typically rely on heuristics that may not yield optimal results, or algorithms that run into issues with scalability and the ability to produce timely results due to the exponential search space. In NM, those problems exist, for instance allocation of resources in case of service function chaining or network slicing among others are recent examples which have gained interest in our community with SDN. Many propositions consist of modeling the optimization problem as an MILP and solve it by means of heuristics to reach a satisfactory tradeoff between solution quality (gap to optimality) - computation time and model size/dimensionality. Hence, ML is recognized to be well adapted to progress on this type of problem [Kaf19].

However, all computational problems of NM are not NP-hard. Due to real-time constraints, some involve very short control loops that require both rapid decisions and the ability to rapidly adapt to new situations and different contexts. So, even in that case, time is critical and approximate solutions are usually more acceptable. Again, it is where AI can be beneficial. Actually, expert systems are AI systems [Ste92] but this kind of systems are not designed to scale with the volume and heterogeneity of data we can collect in a network today for which the expert system is built thanks to numerous inference rules. In contrast, ML is more efficient to automatically learn abstract representations of the rules, which can be eventually updated.

On one hand another type of common problem in NM is classification. For instance, classifying network flows is helpful for security purposes to detect attack flows, to differentiate QoS among the different flows (e.g. real-time streams which need to be prioritized), etc. On the other hand, ML-based classification algorithms have been widely used in literature with high quality results when properly applied leading to their applications in commercial products. There are many algorithms including decision tress, support vector machine or (deep) neural networks which have been to be proven efficient in many areas and notably for image and natural language processing.

Finally, many problems also still rely on humans in the loop, from support issues such as dealing with trouble tickets to planning activities for the roll-out of new services. This creates operational bottlenecks and is often expensive and error prone. This kind of tasks could be either automated or guided by an AI system to avoid human bias. Indeed, the balance between human resources and

the complexity of problems to deal with is actually very imbalanced and this will continue to increase due to the size of networks, heterogeneity of devices, services, etc. Hence, human-based procedures tend to be simple in comparison to the problems to solve or time-consuming. Notable examples are in security where the network operator should defend against potential unknown threat. As a result, services might be largely affected during hours

Actually, all the problems aforementioned are exacerbated by the situation of more complex networks to operate on many dimensions (users, devices, services, connections, etc.). Therefore, AI is expected to enable or simplify the solving of those problems in real networks in the near future [czb20] [Yan20] because those would require reaching unprecedented levels of performance in terms of throughput, latency, mobility, security, etc.

2. Conventions and Definitions

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Acronyms

AI: Artificial Intelligence FL: Federated Learning GAN: Generative Adversarial Network GNN: Graph Neural Network IBN: Intent-Based Networking LSTM: Long Short-Term Memory ML: Machine Learning MLP: Multilayer Perceptron NM: Network Management RL: Reinforcement Learning

4. Difficult problems in network management

As mentioned in introduction, problems to be tackled in NM tend to be complex and exhibit characteristics that make them candidates for solutions that involve AI techniques:

*C1: A very large solution space, combinzatorially exploding with the size of the problem domain. This makes it impractical to explore and test every solution (again NP-hard problems here)

*C2: Uncertainty and unpredictability along multiple dimensions, including the context in which the solution is applied, behavior of users and traffic, lack of visibility into network state, and more. In addition, many networks do not exist in isolation but are subjected to myriads of interdependencies, some outside their control. Accordingly, there are many external parameters that affect the efficiency of the solution to a problem and that

cannot be known in advance: user activity, interconnected networks, etc.

*C3: The need to provide answers (i.e. compute solutions, deliver verdicts, make decisions) in constrained or deterministic time. In many cases, context changes dynamically and decisions need to be made quickly to be of use.

*C4: Data-dependent solutions. To solve a problem accurately, it can be necessary to rely on large volumes of data, having to deal with issues that range from data heterogeneity to incomplete data to general challenges of dealing with high data velocity.

*C5: Need to be integrated with existing automatic and human processes.

*C6: Solutions **MUST** be cost-effective as resources (bandwidth, CPU, human, etc.) can be limited, notably when part of processing is distributed at the network edge or within the network.

Many decision/optimization problems are affected by multiple criteria. Below is a non-exhaustive list of complex NM problems for which AI and/or non-AI-based approaches have been proposed:

*Computation of optimal paths: Packet forwarding is not always based on traditional routing protocols with least cost routing, but on computation of paths that are optimized for certain criteria - for example, to meet certain level objectives, to result in greater resilience, to balance utilization, to optimize energy usage, etc. Many of those solutions can be found in SDN, where a controller or path computation element computes paths that are subsequently provisioned across the network. However, such solutions generally do not scale to millions of paths (C1), and cannot be recomputed in sub-second time scales (C3) to take into account dynamically changing network conditions (C2). To compute those paths, operations research techniques have been extensively used in literature along with AI methods as shown in [Lop20]. As such, this problem can be considered as close to big data problems with some of the different Vs: volume, velocity, variety, value...

*Classification of network traffic: Without loss of generality a common objective of network monitoring for operators is to know the type of traffic going through their networks (web, streaming, gaming, VoIP). By nature, this task analyzes data (C4) which can vary over time (C2) except in very particular scenarios like industrial isolated networks. However, the output of the classification technique is time-constrained only in specific cases where fast decisions **MUST** be made, for example to reroute

traffic. Simple identification based on IANA-assigned TCP/UDP ports numbers were sufficient in the past. However, with applications using dynamic port numbers, signature techniques can be used to match packet payload [Sen04]. To handle applications now encapsulated in encrypted web or VPN traffic, machine-learning has been leveraged [Bri19].

*Network diagnostics: Disruptions of networking services can have many causes and thus can rely on analyzing many sources of data (C4). Identifying the root cause can be of high importance when what is causing the disruption is not properly understood, so that repair actions can address the root cause versus just working around the symptoms. Such repair actions may involve human actions (C5). Further complicating the matter are scenarios in which disruptions are not "hard" but involve only a degradation of service level, and where disruptions are intermittent, not reproducible, and hard to predict. Artificial intelligence techniques can offer promising solutions.

*Intent-Based Networking (IBN): Roughly speaking, IBN refers to the ability to manage networks by articulating desired outcomes without the need to specify a course of actions to achieve those outcomes [RFC9315]. The ability to determine such courses of actions, in particular in scenarios with multiple interdependencies, conflicting goals, large scale, and highly complex and dynamic environments is a huge and largely unsolved challenge (C1, C2, C3). Artificial Intelligence techniques can be of help here in multiple ways, from accurately classifying dynamic context to determine matching actions to reframing the expression of intent as a game that can be played (and won) using artificially intelligent techniques.

*VNF placement and SFC design: Virtual Network Functions need to be placed on physical resources and Service Function Chains designed in an optimized manner to avoid use of networking resources and minimize energy usage (C1,C6).

*Smart admission control to avoid congestion and oversubscription of network resources: Admission control needs to be set up and performed in ways that ensure service levels are optimized in a manner that is fair and aligned with application needs, congestion avoided or its effects mitigated (C6).

5. High-level challenges in adopting AI in NM

As shown in the previous section, AI techniques are good candidates for the difficult NM problems. There have been many propositions but still most of them remain at the level of prototypes or have been only evaluated with simulation and/or emulation. It is thus

questionable why our community investigates much research in this direction but has not adopted those solutions to operate real networks. There are different obstacles.

First, AI advances have been historically driven by the image/video, natural language and signal processing communities as well as robotics for many decades. As a result, the most impressive applications are in this area including recently the generalization of home assistants or the large progress in autonomous vehicles. However, the network experts have been focused on building the Internet, especially building protocols to make the world interconnected and with always better performance and services. This trend continues today with the 5G networks in deployment and beyond 5G under definition. Hence, AI was not the primary focus even if increased network automation calls for AI and ML solutions. However, AI is now considered as a core enabler for the future 6G networks which are sometimes qualified as AI-native networks.

While we can see major contributions in AI-based solutions for networking over more than two decades, only a fraction of the community was concerned by AI at that time. Progress as a whole, from a community perspective, was so limited and compensated by relying on the development of AI in the communities as mentioned earlier. Even if our problems share some commonalities, for example on the volume of data to analyze, there are many differences: data types are completely different, networks are by nature heavily distributed, etc. If problems are different, they **SHOULD** require distinct solutions. In a nutshell, network-tailored AI was overlooked and leads to a first set of challenges described in Section 6.

Second, many AI techniques require enough representative data to be applied independently if the algorithms are supervised or unsupervised. NM has produced a lot of methods and technologies to acquire data. However, in most cases, the goal was not to support AI techniques and lead so to a mismatch. For example, (deep) learning techniques mostly rely on having vectors of (real) numbers as input which fits some metrics (packet/byte counts, latency, delays, etc) but needs some adjustment for categorical (IP addresses, port numbers, etc) or topological features. Conversions are usually applied using common techniques like one-hot encoding or by coarse-grained representations [Sco11]. However, more advanced techniques have been recently proposed to embed representation of network entities rather than pure encoding [Rin17][Evr19][Sol20].

An additional challenge concerns the fact that AI techniques that involve analysis of networking data can also lead to the extraction of sensitive and personally-identifiable information, raising potential privacy concerns and concerns regarding the potential for

abuse. For example, AI techniques used to analyze encrypted network traffic with the legitimate goal to protect the network from intrusions and illegitimate attack traffic could be used to infer information about network usage and interactions of network users. Intelligent data analysis and the need to maintain privacy are in many ways that are contradictory in nature, resulting in an arms race. Similarly, training ML solutions on real network data is in many cases preferable over using less-realisitic synthetic data sets. However, network data may contain private or sensitive data, the sharing of which may be problematic from a privacy standpoint and even result in legal exposure. The challenge concerns thus how to allow AI techniques to perform legitimate network management functions and provide network owners with operational insights into what is going on in their networks, while prohibiting their potential for abuse for other (illegitimate) purposes. Challenges related to network data as input to ML algorithms is detailed in Section 7.

Finally, networks are already operated thanks to (semi-)automated procedures involving a large number of resources which are synchronized with management or orchestration tools. Adding AI supposes it would be seamlessly integrated within pre-existing processes. Although the goal of these procedures might be solely to provide relevant information to operators through alerts or dashboards in case of monitoring applications, many other applications rely on those procedures to trigger actions on the different resources, which can be local or remote. The use of AI or any other approaches to derive NM actions adds further constraint on them, especially regarding time constraints and synchronization to maintain a coherence over a distributed system.

A related challenge concerns the fact that to be deployed, a solution needs to not only provide a technical solution but to also be acceptable to users - in this case, network administrators and operators. One challenge with automated solutions concerns that users want to feel “in control” and able to understand what is going on, even more so if ultimately those users are the ones who are held accountable for whether or not the network is running smoothly. Those same concerns extend to artificially intelligent systems for obvious reasons. To mitigate those concerns, aspects such as the ability to explain actions that are taken - or about to be taken - by AI systems become important.

Beyond reasons of making users more comfortable, there are potentially also legal or regulatory ramifications to ensure that actions taken are properly understood. For example,agencies such as the FCC may impose fines on network operators when services such as E911 experience outages, as there is a public interest in ensuring highest availability for such services. In investigating causes for

such outages, the underlying behavior of systems has to be properly understood, and even more so the reasons for actions that fall under the realm of network operations. All these aspects about integration and acceptability of the integration of AI in NM processes is detailed in Section 8.

6. AI techniques and network management

6.1. Problem type and mapping

In the last few years, an increasing number of different AI techniques have been proposed and applied successfully to a growing variety of different problems in different domains, including network management [Mus18], [Xie18]. Some of the more recently proposed AI approaches are clearly advancements of older approaches, which they supersede. Many other AI approaches are not predecessors or successors but simply complementary because they are useful for different problems or optimize different metrics. In fact, different AI approaches are useful for different kinds of problem inputs (e.g., tabular data vs. text vs. images vs. time series) and also for different kinds of desired outputs (e.g., a predicted value, a classification, or an action). Similarly, there may be trade-offs between multiple approaches that take the same kind of inputs and desired outputs (e.g., in terms of desired objective, computation complexity, constraints).

Overall, it is a key challenge of using AI for network management to properly understand and map which kind of problems with which inputs, outputs, and objectives are best solved with which kind of AI (or non-AI) approaches. Given the wealth of existing and newly released AI approaches, this is far from a trivial task.

6.1.1. Sub-challenge: Suitable Approach for Given Input

Different problems in network management come with widely different problem parameters. For example, security-related problems may have large amounts of text or encrypted data as input, whereas forecasting problems have historical time series data as input. They also vary in the amount of available data.

Both the type and amount of data influences which AI techniques could be useful. On one hand, in scenarios with little data, classical machine learning techniques (e.g., SVM, tree-based approaches, etc.) are often sufficient and even superior to neural networks. On the other hand, neural networks have the advantage of learning complex models from large amounts of data without requiring feature engineering. Here, different neural network architectures are useful for different kinds of problems. The traditional and simplest architecture are (fully connected) multi-layer perceptrons

(MLPs), which are useful for structured, tabular data. For images, videos, or other high-dimensional data with correlation between “close” features, convolutional neural networks (CNNs) are useful. Recurrent neural networks (RNNs), especially LSTMs, and attention-based neural networks (transformers) are great for sequential data like time series or text. Finally, Graph Neural Networks (GNNs) can incorporate and consider the graph-structured input, which is very useful in network management, e.g., to represent the network topology.

The aforementioned rough guidelines can help identify a suitable AI approach and neural network architecture. Still, best results are often only achieved with sophisticated combinations of different approaches. For example, multiple elements can be combined into one architecture, e.g., with both CNNs and LSTMs, and multiple separate AI approaches can be used as an ensemble to combine their strengths. Here, simplifying the mapping from problem type and input to suitable AI approaches and architectures is clearly an open challenge. Future work **SHOULD** address this challenge by providing both clearer guidelines and striving for more general AI approaches that can easily be applied to a large variety of different problem inputs.

6.1.2. Sub-challenge: Suitable Approach for Desired Output

Similar to the challenge of identifying suitable AI approaches for a given problem input, the desired output for a given problem also affects which AI approach **SHOULD** be chosen. Here, the format of the desired output (single value, class, action, etc.), the frequency of these outputs and their meaning **SHOULD** be considered.

Again, there are rough guidelines for identifying a group of suitable AI approaches. For example, if a single value is required (e.g., the amount of resources to allocate to a service instance), then typical supervised regression approaches **SHOULD** be used. If classification (e.g., of malware or another security issue [Abd10]) instead of a value is desired, supervised classification methods **SHOULD** be used. Alternatively, unsupervised machine learning can help to cluster given data into separate groups, which can be useful to analyze networking data, e.g., for better understanding different types of traffic or user segments.

In addition to these classical supervised and unsupervised methods, reinforcement learning approaches allow active, sequential decisions rather than simple predictions or classifications. This is often useful in network management, e.g., to actively control service scaling and placement as well as flow scheduling and routing. Reinforcement learning agents autonomously select suitable actions in a given environment and are especially useful for self-learning

network management. In addition to model-free reinforcement learning, model-based planning approaches (e.g., Monte Carlo Tree Search (MCTS)) also allow choosing suitable actions in a given environment but require full knowledge of the environment dynamics. In contrast, model-free reinforcement learning is ideal for scenarios with unknown environment dynamics, which is often the case in network management.

Similar to the previous sub-challenge, these are just rough guidelines that can help to select a suitable group of AI approaches. Identifying the most suitable approach within the group, e.g., the best out of the many existing reinforcement learning approaches, is still challenging. And, as before, different approaches could be combined to enable even more effective network management (e.g., heuristics + RL, LSTMs + RL, …). Here, further research **MAY** simplify the mapping from desired problem output to choosing or designing a suitable AI approach.

6.1.3. Sub-challenge: Tailoring the AI Approach to the Given Problem

After addressing the two aforementioned sub-challenges, one may have selected a useful kind of AI approach for the given input and output of a network management problem. For example, one may select regression and supervised learning to forecast upcoming network traffic. Or select reinforcement learning to continuously control network and service coordination (scaling, placement, etc.). However, even within each of these fields (regression, reinforcement learning, etc.), there are many possible algorithms and hyperparameters to consider. Selecting a suitable algorithm and parametrizing it with the right hyperparameters is crucial to tailor the AI approach to the given network management problem.

For example, there are many different regression techniques (classical linear, polynomial regression, lasso/ridge regression, SVR, regression trees, neural networks, etc.), each with different benefits and drawbacks and each with its own set of hyperparameters. Choosing a suitable technique depends on the amount and structure of the input data as well as on the desired output. It also depends on the available amount of compute resources and compute time until a prediction is required. If resources and time are not a limiting factor, many hyperparameters can be tuned automatically. In practice, however, the design space of choosing algorithms and hyperparameters is often so large that it cannot be effectively tuned automatically but also requires some initial expertise in selecting suitable AI algorithms and hyperparameters.

This sub-challenge holds for all fields of AI: Supervised learning (regression and classification), self-supervised learning, unsupervised learning, and reinforcement learning, each are broad

and rapidly growing fields. Selecting suitable algorithms and hyperparameters to tailor AI approaches to the network management problem is both an opportunity and a challenge. Here, future work should further explore these trade-offs and provide clearer guidelines on how to navigate these trade-offs for different network management tasks.

6.2. Performance of produced models

From a general point of view, any AI technique will produce results with a certain level of quality. This leads to two inherent questions: (1) what is the definition of the performance in a context of a NM application? (2) How to measure it? and (3) How to ensure/improve the quality of produced results?

Many metrics have been already defined to evaluate the performance of an AI-based techniques in regards to its NM-level objectives. For example, QoS metrics (throughput, latency) can serve to measure the performance of a routing algorithm along with the computational complexity (memory consumption, size of routing tables). The question is to model and measure these two antagonist types of metrics. Number of true/false positives/negatives are the most basic metrics for network attack detection functions. Although the first two questions are thus already answered even if improvement can be done, question (3) refers to the integration of metrics into AI algorithms. Its objective is to obtain the best results which need to be quantified with these metrics. Depending on the type of algorithm, these metrics are either evaluated in an online manner with a feedback loop (for example with reinforcement learning) or in batch to optimize a model based on a particular context (for example described by a dataset for machine learning).

The problem is two-fold. First, the performance can be measured through multiple metrics of different types (numerical or ordinal for example) and some can be constrained by fixed boundaries (like a maximum latency), making their joint use challenging when creating an AI model to resolve a NM problem. Second, the scale metrics differ from each other in terms of importance or impact and can eventually vary on their domains. It can be hard to precisely assess what is a good or bad value (as it might depend on multiple other ones) and it is even more difficult to integrate in an AI technique, especially for learning algorithms to adjust their models based on the performance. Indeed, learning algorithms run through multiple iterations and rely on internal metrics (MAE or (R)MSE for neural network, gini index or entropy for decision trees, distance to an hyperplane for SVMs, etc) which are not strongly correlated to the final metrics of the application. For instance, a decision tree algorithm for classification purposes aims at being able to create branches with a maximum of data from the same classes and so avoid

mixing classes. It is done thanks to a criterion like the entropy index but this kind of Index does not assume any difference between mixing class A and B or A and C. Assuming now that from an operational point of view, if A and B are mixed in the predictions is not critical, the algorithm should have preferred to mix and A and B rather than A and C even if in the first case it will produce more errors.

Therefore, the internal functioning of the AI algorithms should be refined, here by defining a particular criterion to replace the entropy as a quality measure when separating two branches. It assumes that the final NM objectives are integrated at this stage.

Another concrete example is traffic predictors which aim at forecasting traffic demands. They only produce an input that is not necessarily simple to be interpreted and used by, e.g., capacity allocation strategies/policies. A traditional traffic prediction that tries to minimize (perfectly symmetric) MAE/MSE treats positive and negative errors in identical ways, hence is agnostic of the diverse meaning (and costs) of under- and over-provisioning. And, such a prediction does not provide any information on, e.g., how to dimension resources/capacity to accommodate the future demand avoiding all underprovisioning (which entails service disruption) while minimizing overprovisioning (i.e., wasting resources). In other words, it forces the operator to guess the overprovisioning by taking (non-informed) safety margins. A more sensible approach here is instead forecasting directly the needed capacity, rather than the traffic [Beg19].

While the one above is just an example, the high-level challenge is devising forecasting models that minimize the correct objective/loss function for the specific NM task at hand (instead of generic MAE/MSE). In this way, the prediction phase becomes an integral part of the NM, and not just a (limited and hard-to-use) input to it. In ML terms, this maps to solving the loss-metric mismatch in the context of anticipatory NM [Hua19].

Another issue for statistical learning (from examples/observations) is mainly about extracting an estimator from a finite set of input-output samples drawn from an unknown probability distribution that should be descriptive enough for unseen/new input data. In this context online monitoring and error control of the quality/properties of these point estimators (bias, variance, mean squared error, etc.) is critical for dynamic/uncertain network environments. Similar reasoning/challenge applies for interval estimates, i.e., confidence intervals (frequentist) and credible intervals (Bayesian).

6.3. Lightweight AI

Network management and operations often need to be performed under strict time constraints, i.e. at line rate, in particular in the context of autonomic or self-driven networks. Locating NM functions as close as possible where forwarding is achieved is thus an interesting option to avoid additional delays when these operations are performed remotely, for example in a centralized controller. Besides, forwarding devices may offer available resources to supplement or replace edge resources. In case of AI coupled with network management, AI tasks can be offloaded in network devices, or more generally embedded within the network. Obviously, time-critical tasks are the best candidates to be offloaded within the network. Costly learning tasks should be processed in high-end servers but created models can be deployed, configured, modified and tuned in switches.

Recent advances in network programmability ease the programming of specific tasks at data-plane level. P4 [Bos14] is widely used today for many tasks including firewalling [Dat18] or bandwidth management [Che19]. P4 is prone to be agnostic to a specific hardware. Switches actually have particular architectures and the RMT (Reconfigurable Match Table) [Bos13] model is generally accepted to be generic enough to represent limited but essential switch architecture components and functionalities. P4 is inspired by this architecture. The RMT model allows reconfiguring match-action tables where actions can be usual ones (rewrite some headers, forward, drop...). Actions are thus applied on the packets when they are forwarded. Actions can also be more complex programs with some safeguards: no loop, resistivity… The impact on the program development is huge. For example, real number operations are not available by default while they are primordial in many AI algorithms.

In a nutshell, the first challenge to overcome of embedding AI in a network is the capacity of the hardware to support AI operations (architectural limitation). Considering software equipment such as a virtual switch simplifies the problem but does not totally resolve it as, even in that case, strong line-rate requirement limits the type of programs to be executed. For example, BPF (Berkeley Packet Filter) programs provides a higher control on packet processing in OVS [Cha18] but still have some limitations, as the execution time of these programs are bounded by nature to ensure their termination, an essential requirement assuming the run-to-completion model which permits high throughput.

The second challenge (resource limitation) of network-embedded AI in the network is to allocate enough resources for AI tasks with a limited impact on other tasks of network devices such as forwarding, monitoring, filtering… Approximation and/or optimization of AI tasks

are potential directions to help in this area. For instance, many network monitoring proposals rely on sketches and with a proposed well-tuned implementation for data-plane [Liu16][Yan18]. However, no general optimized AI-programmable abstraction exists to fit all cases and proposals are mostly use-case centric. Research direction in NM regarding this issue can benefit from propositions in the field of embedded systems that face the same issues. Binarization of neural networks is one example [Lia18]. Besides, distributed processing is a common technique to distribute the load of a single task between multiple entities. AI task decomposition between network elements, edge servers or controllers has been also proposed [Gup18].

6.4. Distributed AI

Distributed AI assumes different related tasks and components to be distributed across computational resources which are possibly heterogeneous. For example, with advances in transfer and Federated Learning (FL), models can be learned, partially shared and combined or data can be also shared to either improve a local or global model. By nature, a network and a networked infrastructure is distributed and is thus well adapted to any distributed applications. This is exacerbated with the deployment of fog infrastructure mixing network and computational resources. Hence, network management can directly benefit to the distributed network structure to solve its own particular problems but any other type of AI-based distributed applications also assumes communication technologies to enable interactions between the different entities. This leads so the two sub-challenges described hereafter.

6.4.1. Network management for efficient distributed AI

Distributed AI relies on exchanging information between different entities and comes with various requirements in terms of volume, frequency, security, etc. This can be mapped to network requirements such as latency, bandwidth or confidentiality. Therefore, the network needs to provide adequate resources to support the proper execution of the AI distributed application. While this is true for any distributed application, the nature of the problem that is intended to be solved by an AI application and how this would be solved can be considered. For example, with FL, local models can be shared to create a global model. In case of failure of network links or in case of too high latency, some local models might not be appropriately integrated into the global model with a possible impact on AI performance. Depending on the nature of the latter, it might be better to guarantee high performance communications with a few number of nodes or to ensure connectivity between all of them even with lower network performance. Coupling is thus necessary between the network management plane and the distributed AI

applications which leads to a set of questions to be addressed about interfaces, data and information models or protocols. While the network can be adapted or eventually adapt itself to the AI distributed applications, AI applications could also adapt themselves to the underlying network conditions. It paves the way to research on methods to support AI application aware-network management or network-aware AI applications or a mix of both.

6.4.2. Distributed AI for network management

For network management applications relying on distributed AI, challenges from Section 6.4.1 are still valid. Furthermore, network management problems also consider network-specific elements like traffic to be analyzed or configuration to be set on distributed network equipments. Co-locating AI processing and these elements (fully or partially) may help to increase performance. For example, precalculation on traffic data can be offloaded on network routers before being further processed in high-end servers in a data-center. Besides, as data is forwarded through multiple routers, decomposition of AI processes along the forward path is possible [Jos22]. In general, distributed AI-based network management decisions could be made at different nodes in the network based on locally available information [Sch21]. Hence, deployment of AI-based solutions for network management can also take into account various network attributes like network topology, routing policies or network device capability. In that case, management of computational and network resources is even more coupled than in Section 6.4.1 since the network is both part of the AI pipeline resources and the managed object through AI.

A primary application for distributed AI is for management problems that have a local scope. One example concerns problems that can be addressed at the edge, involving tasks and control loops that monitor and apply local optimizations to the edge in isolation from activities conducted by other instances across the network. However, distributed AI can involve techniques in which multiple entities collaborate to solve a global problem. Such solutions lend themselves to problems in which centralized solutions are faced with certain foundational challenges such as security, privacy, and trust: The need to maintain complete state in a centralized solution may not be practical in some cases due to concerns such as privacy and trust among multiple subdomains which may not want to share all of their data even if they would be willing to collaborate on a problem). Other foundational challenges concern issue related to timeliness, in which distributed solutions may have inherent advanges over centralized solutions as they avoid issues related to delays caused by the need to communicate updates globally and across long distances.

6.5. AI for planning of actions

Many tasks in network management revolve around the planning of actions with the purpose of optimizing a network and facilitating the delivery of communication services. For example, Paths need to be planned and set up in ways that minimize wasted network resources (to optimize cost) while facilitating high network utilization (avoiding bottlenecks and the formation of congestion hotspots) and ensuring resiliency (by making sure that backup paths are not congruent with primary paths). Other examples were mentioned in Section 4.

The need for planning only increases with the rise of centralized control planes. The promise of central control is that decisions can be optimized when made with complete knowledge of relevant context, as opposed to distributed control that needs to rely on local decisions being made with incomplete knowledge while incurring higher overhead to replicate relevant state across multiple systems. However, as the scale of networks and interconnected systems continues to grow, so does the size of the planning task. Many problems are NP-hard. As a result, solutions typically need to rely on heuristics and algorithms that often result in suboptimal outcomes and that are challenging to deploy in a scalable manner.

The emergence of Intent-Based Networking emphasizes the need for automated planning even further. The concept underlying “intent” is that it should allow users (network operators, not end users of communication services) to articulate desired outcomes without the need to specify how to achieve those outcomes. An Intent-Based System is responsible for translating the intent into courses of action that achieve the desired outcomes and that continue to maintain the outcomes over time. How the necessary courses of action are derived and what planning needs to take place is left open but where the real challenge lies. Solutions that rely on clever algorithms devised by human developers face the same challenges as any other network management tasks.

These properties (problems with a clearly defined need, whose solution is faced with exploding search spaces and that today rely on algorithms and heuristics that in many cases result only suboptimal outcomes and significant limitations in scale) make automated planning of actions an ideal candidate for the application of AI-based solutions.

AI applications in network management in the past have been largely focusing on classification problems. Examples include analysis by Intrusion Protection Systems of traffic flow patterns to detect suspicious traffic, classification of encrypted traffic for improved QoS treatment based on suspected application type, and prediction of

performance parameters based on observations. In addition, AI has been used for troubleshooting and diagnostics, as well as for automated help and customer support systems. However, AI-based solutions for the automated planning of actions, including the automated identification of courses of action, have to this point not been explored much.

A much-publicized leap in AI has been the development of Alpha Go. Instead of using AI to merely solve classification problems, Alpha Go has been successful in automatically deriving winning strategy for board games, specifically the game of Go which features a prohibitively large search space that was long thought to put the ability to play Go at a world class level beyond the reach of problems that AI could solve. Among the remarkable aspects of Alpha Go is that it is able to identify winning strategies completely on its own, without needing those strategies to be taught or learned by observations assuming the system is aware of rules.

The challenge for AI in network management is hence, where is the equivalent of an Alpha Go that can be applied to network management (and networking) problems? Specifically, better solutions are needed for solutions that automatically derive plans and courses of actions for network optimization and similar NP-hard problems, such as provided today with only limited effectiveness by controllers and management applications.

Also, the evaluation of AI algorithms to derive courses of actions is more complex than more common regression or classification tasks. Actions need to be applied in order to observe the results it leads to. However, contrary to game playing, solutions need to be applied in the real world, where actions have real effects and consequences. Different orientations can be envisioned. First, incremental application of AI decisions with small steps can allow us to carefully observe and detect unexpected effects. This can be complemented with roll-back techniques. Second, formal verification techniques can be leveraged to verify decisions made by AI are maintained within safety bounds. Third, sandbox environments can be used but they **SHOULD** be representative enough of the real world. After progress in simulation and emulation, recent research advances lead to the definition of digital twins which implies a tight coupling between a real system and its digital twin to ensure a parallel but synchronized execution. Alternatively, transfer learning techniques in another promising area to be able to capitalize on ML models applicable on a real word system in a more generic sandbox environment. It is actually also an open problem to make the use of AI more acceptable as highlighted in the dedicated section.

7. Network data as input for ML algorithms

Many applications of AI takes as input data. The quality of the outputs of ML-based techniques are highly dependent on the quality and quantity of data used for learning but also on other parameters. For example, as modern network infrastructures move towards higher speed and scale, they aim to support increasingly more demanding services with strict performance guarantees. These often require resource reconfigurations at run time, in response to emerging network events, so that they can ensure reliable delivery at the expected performance level. Timely observation and detection of events is also of paramount importance for security purposes, and can allow faster execution of remedy actions thus leading to reduced service downtime.

Thus, the challenge of data management is multifaceted as detailed in next subsections.

7.1. Data for AI-based NM solutions

Assuming a network management application, the first problem to address is to define the data to be collected which will be appropriate to obtain accurate results. This data selection can require defining problem-specific data or features (feature engineering).

Firstly, NM has already produced a lot of methods and technologies to acquire data. However, in most cases, the goal was not to support AI problems and lead to a mismatch. Indeed, machine learning algorithms only work as desired when data to be analyzed respects properties. Many methods rely on vector-based distances which so supposes that the data encoded into the vector respects the underlying distance semantic. Taking the first n bytes of a packet as vectors and computing distances accordingly is possible but does not embed the semantic of the information carried out in the headers. For example, (deep) learning techniques mostly rely on vectors of (real) numbers as input which fits some metrics (packet/ byte counts, latency, delays, etc) but needs some adjustment for categorical (IP addresses, port numbers, etc) or topological features. Conversions are usually applied using common techniques like one-hot encoding or by coarse-grained representations [Sco11]. However, more advanced techniques have been recently proposed to embed representation of network entities rather than pure encoding [Rin17][Evr19][Sol20]. Data to handle can be in a schema-free or eventually text-based format. One example could be the automated annotation of management intents provided in an unstructured textual format (policies descriptions, specifications,) to extract from them management entities and operations. For that purpose, suitable annotation models need to be built using existing NER (Named Entity

Recognition) techniques usually applied for NLP. However, this **SHALL** be carefully crafted or specialized for network management (intent) language which indirectly bounces back to the challenges of AI techniques for NM specified earlier.

Secondly, The behavior of any network is not just derived from the events that can be directly observed, such as network traffic overload, but also from events occurring outside the environment of the network. The information provided by the detectors of such kinds of events, e.g. a natural incident (earthquake, storm), can be used to determine the adaptation of the network to avoid potential problems derived from such events. Those can be provided by BigData sources as well as sensors of many kinds. The AI challenge related to this task is to process large amounts of data and associate it with the effects that those events have on the network. It is hard to determine the static and dynamic relation between the data provided by external sources and the specific implications it has in networks. For instance, the effect of a “flash crowd” detected in an external source depends on the relation of a particular network to such an event. This can be addressed by AI and its particular application to network management. The objective is to complement a control-loop, as shown in [Mar18], by including the specific AI engines into the decision components as well as the processes that close the loop, so the AI engine can receive feedback from the network in order to improve its own behavior. Similar challenges are addressed in other domains, image processing and computer vision, by using artifacts for anticipating movements in object location and identification.

7.2. Data collection

Once defined, the second problem to address is the collection of data. Monitoring frameworks have been developed for many years such as IPFIX [RFC7011] and more recently with SDN-based monitoring solutions [Yu14][Ngu20]. However, going towards more AI for actions in network management supposes also to retrieve more than traffic related information. Actually, configuration information such as topologies, routing tables or security policies have been proven to be relevant in specific scenarios. As a result, many different technologies can be used to retrieve meaningful data. To support improved QoE, monitoring of the application layer is helpful but far from being easy with the heterogeneity of end-user applications and the wide use of encrypted channels. Monitoring techniques need to be reinvented through the definition of new techniques to extract knowledge from raw measurement [Bri19] or by involving end-users with crowd-sourcing [Hir15] and distributed monitoring. Also, the data-mesh concept proposes to classify data into three categories: source-aligned, aggregate and consumer-aligned. Source-aligned data are those related to the same operational domain and it is important

to correlate or aggregate them with higher planes: management-, control- and forwarding plane. An issue is the difference, not only in the nature of data, but in their volumes and their variety. Some may change rapidly over time (for example network traffic) while other may be quite stable (device state).

The collecting process requirements depend on the kind of processing. We can distinguish two major classes: batch/offline vs real-time/online processing. In particular, real-time monitoring tools are key in enabling dynamic resource management functions to operate on short reconfiguration cycles. However, maintaining an accurate view of the network state requires a vast amount of information to be collected and processed. While efficient mechanisms that extract raw measurement data at line rate have been recently developed, the processing of collected data is still a costly operation. This involves potentially sampling, evaluating and aggregating a vast amount of state information as a response to a diverse set of monitoring queries, before generating accurate reports. One difficult problem resides also in the availability of data as real-time data from different sources to be aggregated may not arrive at the same time requiring so some buffering techniques. Machine learning methods, e.g. based on regression, can be used to intelligently filter the raw measurements and thus reduce the volume of data to process. For example, in [Tan20] the authors proposed an approach in which the classifiers derived for this purpose (according to measurements on traffic properties) can achieve a threefold improvement in the query processing capability. A residual question is the storage of raw measurements. In fact, predicting the lifetime of data is challenging because their analysis may not be planned and triggered by a particular event (for example, an anomaly or attack). As a result, the provisioning of storage capacity can be hard.

In parallel to the continuously increasing dynamicity of networks and complexity of traffic, there is a trend towards more user traffic processing customization [RFC8986][Li19]. As a result, fine grained information about network element states is expected and new propositions have emerged to collect on-path data or in-band network telemetry information [Tan20b]. These new approaches have been designed by introducing much flexibility and customization and could be helpful to be used in conjunction with AI applications. However, the seamless coupling of telemetry processes with packet forwarding requires careful definition of solutions to limit the overhead and the impact of the throughput while providing the necessary level of details. This shares commonalities with the lightweight AI challenge.

7.3. Usable data

Although all agree on the necessity to have more shared datasets, it is quite uncommon in practice. Data contains private or sensitive information and may not be shared because of the criticality of data (which can be used by ill-intentioned adversaries) or due to laws or regulations, even within the same company. To solve this issue, anonymization techniques [Dij19] can be enhanced to optimize the trade-off between valuable data vs sensitive information (potential) leakage or reconstruction. Whatever the final user of data, regulations and laws impose rules on data management with potentially costly impact if they are not respected voluntarily or not. Defining a new monitoring framework should always consider security and privacy aspects, for example to let any user/customer or access/remove its own data with General Data Protection Regulation (GDPR) in EU. The challenge resides here in the capacity of qualifying what is critical or private information and the capacity for an adversary to reconstruct it from other sources of data. Hence AI/ML based solutions will require more data but also more administrative, legal and ethical procedures. Those can last long and so slow down the deployment of a new solution. In addition, this requires interaction with experts from different domains (e.g. AI engineer and a lawyer). The integration of these non-technical constraints should be considered when defining new data to be collected or a new technique to collect data. However, knowing the final use of data is most of the time necessary for ethical and legal assessment which assumes that those considerations **SHOULD** be integrated from the early design of new AI-based solutions.

For supervised or semi-supervised training, having a labeled dataset is a prerequisite. It constitutes a major challenge as well. One one hand, collectors are able to retrieve data. On the other hand, those network data are typically unlabeled. This limits application of ML to unsupervised learning tasks (learning from data). Because manual labeling is a tedious task. one option is to leverage AI to guide humans. This may also support a better generalization of a learned model. Indeed, an underlying challenge is the genericity or coverage of the datasets. Labels encode values of an objective function, the challenge posed by the design of such tools is tremendous since for involving a M:N relationship: 1 data type may be associated to M objective function values and N data types may be associated to 1 objective function. As a result, most datasets used for research encodes a single label for a particular application like attack label for datasets to be used in the context of intrusion detection or application type for network traffic used for classification where the value of a single dataset could be capitalized in several applications.

Again, researchers need empirical (or at least realistic) datasets to validate their solutions. Unfortunately, as highlighted above, having such data from real deployments for various reasons (business secrets, privacy concerns, concerns that vulnerabilities are revealed by accident, raw unlabeled data, etc.) is tough. Even if such a dataset is available it might not be enough to convincingly validate a new algorithm. Instead of falling back to artificial testbed experiments or simulation, it would be useful to have the capability to generate datasets with characteristics that are not 100% identical but similar to the characteristics of one or more real datasets. Such synthetic networks can be used to validate new management algorithms, intrusion detection systems, etc. The usage of AI (for example GANs) in this area [Hui22] is not yet widespread and there are still many concerns that deter researchers, e.g. the fear of leaking sensitive information from the original dataset into the synthetic dataset.

8. Acceptability of AI

Networks are critical infrastructures. On one hand, they **SHOULD** be operated without interruption and must be interoperable. Networks, except in a lab, are not isolated which slow down innovation in general. For example, changing Internet routing protocols **SHOULD** be accepted by all. The same applies for protocol. Even if there have been several versions of major protocols in use like TCP or DNS, there are still some security issues which cannot be patched with 100% guarantee. On the other hand, results provided by AI solutions are uncertain by nature. The same technique applied in different environments can produce different results. AI techniques need some effort (time and human) to be properly configured or to be stabilized. For instance, reinforcement learning needs several iterations before being able to produce acceptable results. These properties of AI techniques are thus a bit antagonist with the criticality of network infrastructures. With that in mind, acceptability of AI by network operators is clearly an obstacle for its larger adoption.

8.1. Explainability of Network-AI products

A common issue across all Machine Learning (ML) applications is that they are black boxes. This means that, after training, the knowledge acquired by ML models is unintelligible to humans. As a result, offering hard guarantees on performance is a very challenging issue. In addition, complex ML models like neural networks -that often have more than hundreds of thousands of parameters- are very hard to debug or troubleshoot in case of failure.

While this is a common issue for all applications of AI, many areas work well with uncertainty and the black-box behavior of AI-based

solutions. For instance, users accept an inherent error in recommender systems or computer vision solutions.

The networking field has already produced a set of well-established network management algorithms and methods, with clear performance guarantees and troubleshooting mechanisms [Rex06][Kr14]. As such, improving debugging, troubleshooting and guarantees on AI-based solutions for networking is a must.

AI researchers and practitioners are devoting large research efforts to improve this aspect of ML models, which is commonly known as explainability [XAI].

This set of techniques provides insights and, in some cases, guarantees on the performance and behavior of ML-based solutions. Understanding such techniques, researching and applying them to network AI is critical for the success of the field.

There exist several ML-based methods that are human-understandable, although not widely used today. For instance, [Mar20] shows a method for building anticipation models (prediction) that provide explanations while determining some actions for tuning some parameters of the network. There are other challenges that **SHOULD** be addressed, such as providing explanations for other ML methods that are quite extended. For instance, xNN/SVM models can be accompanied by Digital Twins of the network that are reversely explored to explain some output from the ML model (e.g., xNN/SVM). In this context, there already exist several methods [Zil20][Puj21] that produce human-readable interpretations of trained NN models, by analyzing their neural activations on different inputs. (As an aside, it should be noted that Digital Twins are not considered per se an AI approach; they merely serve to provide a digital representation of a network that can serve as its proxy and offer a layer of indirection between management applications and actual network resources. That being said, it is conceivable that AI-based management applications can be combined and operate in conjunction with Digital Twin technology, for example to use a Digital Twins as an experimentation sandbox or staging ground for AI-driven applciations.)

8.2. AI-based products and algorithms in production systems

AI-based network management and optimization algorithms are first trained, then the resulting model is used to produce relevant inferences in operation, either in management or optimization scenarios. A relevant question for the success of AI-based solutions is: where does this training occur?

Traditionally, AI-based models have been trained in the same scenario where they operate[Val17][Xu18], this is the customer network. However this presents critical drawbacks. First, training an AI model for management and operation typically requires generating network configurations and scenarios that can break the network. This is because training requires seeing a broad spectrum of scenarios. Thus, it is not feasible in production networks. Second, customer networks may not be equipped with the monitoring infrastructure required to collect the data used in the training process (e.g., performance metrics).

A more sensible approach is to train the AI-based product in a lab, for instance in the vendor's premises. In the lab, AI models can be trained in a controlled testbed, with any configuration, even ones that break the network. However, the main challenge here arises from the fundamental differences between the lab's network and the customer networks. For instance, the topology of the lab's network might be smaller, etc. As a result, there is a need for models that are able to generalize. In this context, generalization means that models should be able to operate in other scenarios not seen during training, with different topologies, routing configurations, scheduling policies, etc.

In order to address this generalization problem, multiple complementary approaches are possible: One approach is training on diverse data that represents large parts of the expected problem space. For example, training with various different traffic patterns will help improve generalization to unseen but comparable traffic patterns. Another approach is to leverage AI designs or architectures that facilitate generalization. One example are Graph Neural Networks (GNN) [gnn1][gnn2]. GNNs are a rather novel type of neural network able to operate and generalize over graphs. Indeed, networks are fundamentally represented as graphs: topology, routing, etc. With GNNs, vendors can train the AI model in a lab with a certain topology and then directly use the resulting model in different customer networks, even with different network topologies. Finally, another approach is Transfer Learning [tl1]. With this technique, the knowledge gained in the lab's training is used to operate in the customer network. Transfer Learning still requires that some data from the customer is used to re-train and fine-tune the model (e.g., accurate performance measurements). This means that, for each customer network, re-training is required. This may be problematic since it requires added cost and access to customer data.

In addition to the challenge of generalizing from training to production environment, there are also challenges in terms of interoperability between different AI approaches and different deployment environments. As mentioned above, AI approaches may be

deployed in diverse environments, e.g., for training and production, but also for local development, for testing, and for validation or in different part of the production systems. These environments may differ in available compute resources, network topology, operating systems, cloud providers, etc. (single node machine, single cluster, many distributed clusters, ...). Deploying the same AI solutions in these different environments can lead to various challenges in terms of interoperability. Common AI frameworks support scaling across networks of different size. Yet, many frameworks are often combined, e.g., for data collection, processing, predictions, validation, etc. Again, ensuring interoperability between these frameworks can be tedious.

This shares some with problems described in Section 6.4 and particularly emphasizes the need for network environments to provide interfaces and descriptions suitable for AI solutions to be properly instantiated and configured.

One approach to address these interoperability challenges is through meta-frameworks that interface with most available AI frameworks. These meta-frameworks provide a higher level of abstraction and often allow seamless deployment across different environments (e.g., on-premise or at different cloud providers) [Mor18].

8.3. AI with humans in the loop

Depending on the network management task, AI can automate and replace manual human control or it can complement human experts and keep them in the loop. Keeping humans in the loop will be an important step of building trust in AI approaches and help ensure the desired outcomes. There are various ways of keeping humans in the loop in the different fields of AI, which could be useful for different aspects of network management.

In classification tasks (e.g., detecting security breaches, malware or detecting anomalies), trained AI models provide a confidence score in addition to the predicted class. If the confidence is high, the prediction is used directly. If the confidence is too low, a human expert may jump in and make the decision - thereby also providing valuable training data to improve the AI model. Such approaches are already being used in industry, e.g., to automatically label datasets (AWS SageMake). Similar approaches could also be used for other supervised learning tasks, e.g., regression. Still, it is an open challenge to keep humans in the loop in all phases of the learning process.

Another field of AI is reinforcement learning, which is useful for taking continuous control decisions in network management, e.g., controlling service scaling and placement as well as flow scheduling

and routing over time. Reinforcement learning agents typically interact with the environment (i.e., the simulated or real network) completely autonomously without human feedback. However, there is a growing number of approaches to put human experts back into the loop. One approach is offline reinforcement learning, where the training data does not come from the reinforcement learning agent's own exploration but from pre-recorded traces of human experts (e.g., placement decisions that were made by humans before). Another approach is to reward the reinforcement learning agent based on human feedback rather than a pre-defined reward function [Lee21]. Again, while there are first promising approaches, more work is required in this area. Overall, it is an open challenge to both leverage the benefits of AI but keep human experts in the loop where it is useful.

9. Security Considerations

TODO Security

10. IANA Considerations

This document has no IANA actions.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/info/rfc7011>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <https://www.rfc-editor.org/info/rfc8986>.

[RFC9315] Clemm, A., Ciavaglia, L., Granville, L. Z., and J. Tantsura, "Intent-Based Networking - Concepts and

Definitions", RFC 9315, DOI 10.17487/RFC9315, October 2022, <https://www.rfc-editor.org/info/rfc9315>.

11.2. Informative References

[Abd10] Jalil, K. A., Kamarudin, M. H., and M. N. Masrek, "A Diagnosis Expert System for Network Traffic Management", 2010. IEEE international conference on networking and information technology

[Beg19] Bega, D., Gramaglia, M., Fiore, M., Banchs, A., and X. Costa-Perez, "DeepCog: Cognitive Network Management in Sliced 5G Networks with Deep Learning", 2019. IEEE INFOCOM

[Bos13] Bosshart, P., Gibb, G., Kim, H.-S., Varghese, G., McKeown, N., Izzard, M., Mujica, F., and M. Horowitz, "Forwarding metamorphosis: Fast programmable match-action processing in hardware for SDN", 2013. ACM SIGCOMM

[Bos14] Bosshart, P., Daly, D., Gibb, G., Izzard-, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., and D. Walker, "P4: programming protocol-independent packet processors", 2014. SIGCOMM Comput. Commun. Rev. 44

[Bou18] Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities", 2018. Journal of Internet Services and Applications 9, 16

[Bri19] Brissaud, P.-O., François, J., Chrisment, I., Cholez, T., and O. Bettan, "Transparent and Service-Agnostic Monitoring of Encrypted Web Traffic", 2019. IEEE Transactions on Network and Service Management, 16 (3)

[Cha18] Chaignon, P., Lazri, K., François, J., Delmas, T., and O. Festor, "Oko: Extending Open vSwitch with Stateful Filters", 2018. ACM Symposium on SDN Research (SOSR)

[Che19] Chen, Y., Yen, L., Wang, W., Chuang, C., Liu, Y., and C. Tseng, "P4-Enabled Bandwidth Management", 2019. Asia-Pacific Network Operations and Management Symposium (APNOMS)

[czb20] Clemm, A., Zhani, M. F., and R. Boutaba, "Network Management 2030: Operations and Control of Network 2030

Services", 2020. Springer Journal of Network and Systems Management (JNSM)

[Dat18] Datta, R., Choi, S., Chowdhary, A., and Y. Park,, "P4Guard: Designing P4 Based Firewall", 2018. IEEE Military Communications Conference (MILCOM)

[Dij19] Dijkhuizen, N. V., Ham, J. V. D., and X. Li, "A Survey of Network Traffic Anonymisation Techniques and Implementations", 2014. ACM Comput. Surv. 51, 3, Article 52

[Evr19] Evrard, L., François, J., Colin, J.-N., and F. Beck, "port2dist: Semantic Port Distances for Network Analytics", 2019. IFIP/IEEE Symposium on Integrated Network and Service Management (IM)

[gnn1] Battaglia, P. W. and E. al, "Relational inductive biases, deep learning, and graph networks", 2018. arXiv preprint arXiv:1806.01261

[gnn2] Rusek, K., Suárez-Varela, J., Mestres, A., Barlet-Ros, P., and A. Cabellos-Aparicio, "Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN", 2019. ACM Symposium on SDN Research

[Gup18] Gupta, A., Harrison, R., Canini, M., Feamster, N., Rexford, J., and W. Willinger, "Sonata: query-driven streaming network telemetry", 2018. ACM SIGCOMM Conference

[Hir15] Hirth, M., Hossfeld, T., Mellia, M., Schwartz, C., and F. Lehrieder, "Crowdsourced network measurements: Benefits and best practices", 2015. Computer Networks. 90

[Hoo18]

Hooft, J. V. D., Claeys, M., Bouten, N., Wauters, T., Schönwälder, J., Stiller, A. P. B., Charalambides, M., Badonnel, R., Serrat, J., Santos, C. R. P. D., and F. D. Turck, "Updated Taxonomy for the Network and Service Management Research Field", 2018. Journal of Network System Managemen (JNSM) 26, 790-808

[Hua19] Huang, C., Zhai, S., Talbott, W., Bautista, M. A., Sun, S.-Y., Guestrin, C., and J. Susskind, "Addressing the

Loss-Metric Mismatch with Adaptive Loss Alignment", 2020. ICRL

[Hui22] Hui, S., Wang, H., Wang, Z., Yang, X., Liu, Z., Jin, D., and Y. Li, "Knowledge Enhanced GAN for IoT Traffic Generation", 2022. ACM Web Conference 2022 (WWW)

[Jos22] Jose, M., Lazri, K., François, J., and O. Festor, "NetREC Network-wide in-network REal-value Computation.", 2022. IEEE International Conference on Network Softwarization (NetSoft)

[Kaf19] Kafle, V. P., Martinez-Julia, P., and T. Miyazawa, "Automation of 5G Network Slice Control Functions with Machine Learning", 2019. IEEE Communications Standards Magazine, vol. 3, no. 3, pp. 54-62

[Kr14] Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., and S. Uhlig, "Software-defined networking: A comprehensive survey", 2015. Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76

[Lee21] Lee, K., Smith, L., and P. Abbeel, "Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training", 2021. arXiv preprint arXiv:2106.05091

[Li19] Li, R., Makhijani, K., Yousefi, H., Westphal, C., Dong, L., Wauters, T., and F. D. Turck., "A Framework for Qualitative Communications Using Big Packet Protocol", 2019. ACM SIGCOMM Workshop on Networking for Emerging Applications and Technologies (NEAT)

[Lia18] Liang, S., Yin, S., Liu, L., Luk, W., and S. Wei, "FP-BNN: Binarized neural network on FPGA", 2018. Neurocomputing, Volume 275

[Liu16] Liu, Z., Manousis, A., Vorsanger, G., Sekar, V., and V. Braverman, "One Sketch to Rule Them All: Rethinking Network Flow Monitoring with UnivMon", 2016. ACM SIGCOMM Conference

[Lop20] López, J., Labonne, M., Poletti, C., and D. Belabed, "Priority Flow Admission and Routing in SDN: Exact and Heuristic Approaches", 2020. IEEE International Symposium on Network Computing and Applications (NCA)

[Mar18] Martinez-Julia, P., Kafle, V. P., and H. Harai, "Exploiting External Events for Resource Adaptation in Virtual Computer and Network Systems", 2018. IEEE

Transactions on Network and Service Management, Vol. 15, N. 2,

[Mar20] Martinez-Julia, P., Kafle, V. P., and H. Asaeda, "Explained Intelligent Management Decisions in Virtual Networks and Network Slices", 2020. Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)

[Mor18] Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M., and I. Stoica, "Ray: A Distributed Framework for Emerging AI Applications", 2018. USENIX Symposium on Operating Systems Design and Implementation (OSDI)

[Mus18] Musumeci, F., Rottondi, C., Nag, A., Macaluso, I., Zibar, D., Ruffini, M., and M. Tornatore, "An overview on application of machine learning techniques in optical networks", 2018. IEEE Communications Surveys & Tutorials, 21(2), 1383-1408.

[Ngu20] Nguyen, T. G., Phan, T. V., Hoang, D. T., Nguyen, T. N., and C. So-In, "Efficient SDN-based traffic monitoring in IoT networks with double deep Q-network", 2020. International conference on computational data and social networks, Springer

[Puj21] Pujol-Perich, D., Suárez-Varela, J., Xiao, S., Wu, B., Cabello, A., and P. Barlet-Ros, "NetXplain: Real-time explainability of Graph Neural Networks applied to Computer Networks", 2021. MLSys workshop on Graph Neural Networks and Systems (GNNSys)

[Rex06] Rexford, J., "Route optimization in IP networks", 2006. Handbook of Optimization in Telecommunications (pp. 679-700), Springer

[Rin17] Ring, M., Dallmann, A., Landes, D., and A. Hotho, "IP2Vec: Learning Similarities Between IP Addresses", 2017. IEEE International Conference on Data Mining Workshops (ICDMW)

[Sch21] Schneider, S., Qarawlus, H., and H. Karl, "Distributed Online Service Coordination Using Deep Reinforcement

Learning", 2021. IEEE International Conference on Distributed Computing Systems (ICDCS)

[Sco11] Coull, S. E., Monrose, F., and M. Bailey, "On Measuring the Similarity of Network Hosts: Pitfalls, New Metrics, and Empirical Analyses", 2011. NDSS

[Sen04] Sen, S., Spatscheck, O., and D. Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures", 2004. ACM International conference on World Wide Web (WWW)

[Sol20] Soliman, H. M., Salmon, G., Sovilij, D., and M. Rao, "A Graph Neural Network Approach for Scalable and Dynamic IP Similarity in Enterprise Networks", 2020. IEEE International Conference on Cloud Networking (CloudNet)

[Ste92] Stern, D. and P. Chemouil, "A Diagnosis Expert System for Network Traffic Management", 1992. Networks, Kobe, Japan

[Tan20] Tangari, G., Charalambides, M., Pavlou, G., Grazian, C., and D. Tuncer, "Classification-assisted Query Processing for Network Telemetry", 2020. Network Traffic Measurement and Analysis Conference (TMA)

[Tan20b] Lizhuang, T., Wei, S., Zhenyi, Z., Jingying, M., Xiaoxi, L., and L. Na, "In-band Network Telemetry: A Survey", 2020. Computer Networks. 186. 10.1016

[tl1] Torrey, L. and J. Shavlik, "Transfer learning", 2010. Handbook of research on machine learning applications and trends: algorithms, methods, and techniques

[Val17] A., V., M., S., D., S., and T. A., "Learning to route", 2017. ACM HotNets

[XAI] Samek, W., Wiegand, T., and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models", 2017. arXiv preprint arXiv:1708.08296

[Xie18] Xie, J., Yu, F. R., Huang, T., Xie, R., Liu, J., Wang, C., and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges", 2018. IEEE Communications Surveys & Tutorials

[Xu18] Z., X., J., T., J., M., W., Z., Y., W., H., L. C., and Y. D., "Experience-driven networking: A deep reinforcement learning based approach", 2018. IEEE INFOCOM

[Yan18] Yang, T., Jiang, J., Liu, P., Huang, Q., Gong, J., Zhou, Y., Miao, R., Li, X., and S. Uhlig, "Elastic sketch: adaptive and fast network-wide measurements", 2018. ACM SIGCOMM Conference

[Yan20] Yang, H., Alphones, A., Xiong, Z., Niyato, D., Zhao, J., and K. Wu,, "Artificial-Intelligence-Enabled Intelligent 6G Networks", 2020. IEEE Network, vol. 34, no. 6, pp. 272-280

[Yu14] Yu, Y., Qian, C., and X. Li, "Distributed and collaborative traffic monitoring in software defined networks", 2014. ACM Hot topics in software defined networking

[Zil20] Meng, Z., Wang, M., Bai, J., Xu, M., Mao, H., and H. Hu, "Interpreting Deep Learning-Based Networking Systems", 2020. ACM SIGCOMM

Acknowledgments

This document is the result of a collective work. Authors of this document are the main contributors and the editors but contributions have been also received from the following people we acknowledge: Laurent Ciavaglia, Felipe Alencar Lopes, Abdelkader Lahamdi, Albert Cabellos, José Suárez-Varela, Marinos Charalambides, Ramin Sadre, Pedro Martinez-Julia and Flavio Esposito

This document is also partially supported by project AI@EDGE, funded from the European Union's Horizon 2020 H2020-ICT-52 call for projects, under grant agreement no. 101015922.

The views expressed in this document do not necessarily reflect those of the Bank of Canada's Governing Council

Authors' Addresses

Jérôme François
University of Luxembourg and Inria
6 Rue Richard Coudenhove-Kalergi
L- Luxembourg
Luxembourg

Email: jerome.francois@uni.lu

Alexander Clemm
Futurewei Technologies, Inc.
United States of America

Email: ludwig@clemm.org

Dimitri Papadimitriou
3NLab Belgium Reseach Center
Leuven
Belgium

Email: papadimitriou.dimitri.be@gmail.com

Stenio Fernandes
Central Bank of Canada
Canada

Email: stenio.fernandes@ieee.org

Stefan Schneider
Digital Railway (DSD) at Deutsche Bahn
Germany

Email: stefanbschneider@outlook.com