NSF Workshop on Measurements for Self-Driving Networks

Nick Feamster
Princeton University

Arpit Gupta
UC Santa Barbara

Jennifer Rexford
Princeton University

Walter Willinger
NIKSUN, Inc.

April 4–5, 2019

Contents

1 Background

Networking researchers, equipment vendors, and Internet service providers alike have argued for developing more autonomous or self-managing networks in the past. However, the recent proliferation of networked devices (*e.g.*, Internet-of-Things) and systems (*e.g.*, container networks) as well as distributed applications (*e.g.*, e-commerce) has brought to the fore the limitations of conventional network-management tools and systems, especially their inability to deal with the growing complexity of today's networks. At the same time, this pull for more autonomous networks has been accompanied by a recent technological push that has witnessed (1) the development of fully programmable, protocol-independent data planes and languages for programming them; and (2) the emergence of new platforms, tools, and algorithms for Artificial Intelligence (AI) and Machine Learning (ML).

Moreover, today's large cloud, content, and Internet service providers (*e.g.*, Amazon, Google, Verizon, etc.) are also experiencing significant economic pressure to operate their networks in ever-more efficient and cost-effective ways. However, as the complexity of network management continues to increase and, at the same time, the time scale to detect and react to different types of network events continues to decrease, these providers have reached a point where many of the tasks can no longer be performed by human network operators. To minimize the dependence on human operators and thus also save costs, network operators are therefore increasingly trying to automate more and more of their day-to-day network operations. In fact, most modern provider networks view network automation no longer as a luxury but as absolutely critical for their business. These and related developments have further strengthened the academic and industry researchers' recent resolve to revive the idea of autonomous networks; *i.e.*, "networks that can run (drive) by themselves".

In response to this renewed interest within academia and industry in leveraging the latest technologies to advance the development of autonomous networks, some of us (Nick Feamster and Jennifer Rexford) organized the NSF-sponsored **Workshop on Self-Driving Networks** that was held at Princeton University on February 15-16, 2018. The workshop focused on outlining a few use cases for self-driving networks and identifying related research problems. In its conclusion, the workshop report stated that *"making progress will rely on interdisciplinary collaboration between computer networking and other areas (e.g., machine learning, distributed systems, security, and programming languages), as well as stronger connections with operational networks to acquire data, identify important use cases, and evaluate solutions."* For a summary of the discussions that took place during the workshop, see the published workshop report [9].

2 Workshop Overview

Building on the success of this 2018 workshop and pursuing the long-term goal of realizing the vision outlined in [9] (*i.e.*, *" that networks might one day be able to largely manage themselves, through a combination of query-driven network measurement, automated inference techniques, and programmatic control"*), we decided to organize a follow-up workshop in 2019 that picked up where the 2018 workshop left off. The NSF-sponsored **Workshop on Measurements for Self-Driving Networks** was held at Princeton University on April 4-5, 2019, and was mainly concerned with issues that the 2018 workshop identifies as being critical for sustained progress in this area of research—leveraging real-world production networks for both acquiring data (for developing new AI/ML-based models in support of automated inference) and evaluating solutions (for "road-testing" the developed prototypes) in support of self-driving networks.

Noting that in the past, AI/ML applications have excelled in areas that had an abundance of (labeled) data, it is natural to expect that the availability of data will similarly play a critical role in fueling the development of self-driving networks. Importantly, to ensure practical relevance and exposure to realistic conditions, the data of interest should be collected from real-world production networks. However, due to the scale and privacy-sensitivity of the resulting data, making it available for today's researchers is often problematic, if not impossible. Existing tools and techniques are not designed to collect and process privacy-sensitive network data for self-driving networks. To discuss these and related issues, this workshop brought together experts interested in different aspects of data-driven research on self-driving networks, including:

- The collection of diverse and high-quality data from real-world production networks for training and evaluating new learning models specifically designed to automatically detect and diagnose different types of network events with high confidence,

- Real-time or post-facto processing of parts of the network data in a privacy-preserving manner to ensure data privacy or protect commercially sensitive information,
- The design of in-network measurement architectures for adapting the evaluated training models in real time to make them robust to the uncertainty in the environment, and
- Access to and use of testbeds and real-world production networks for examining the relevance and utility of proposed tools and prototypes in realistic settings with minimal disruptions of normal network operations.

To make best use of the diverse backgrounds and different areas of expertise of the workshop attendees, we organized the workshop into four sessions. Each session started out with 1-2 talks that set the stage and were followed by a breakout session. In an attempt to establish stronger connections with operational network providers, a number of the invited speakers were network operators who have been applying AI/ML to automate various management tasks in their networks. In addition, we also divided each of the breakout sessions into two groups. While the first group focused on issues related to managing campus, home, broadband, enterprise, and mobile networks, the second group discussed topics that concerned the management of the infrastructures and services of the large cloud/content provider networks as well as the operations of intra-data center and wide-area networks. Our reason for this division of the breakout sessions was to identify fundamental research challenges that need to be addressed across eyeball and cloud/content networks and elaborate on common design patterns that can be leveraged when developing self-driving networks.

In the course of this workshop, we also noticed that while there is a consensus on the need to design self-driving networks, the question of what precisely self-driving networks are or should be is less clear. In fact, given the diversity in applications, devices, systems, infrastructure support, etc., formalizing the requirements and defining success for building self-driving networks remains mostly an open question. However, we believe that just as the development of advanced driver assistant systems (*e.g.*, adaptive cruise control, lane departure warning, etc.) is a critical first step towards the development of self-driving cars, network automation plays an equally important role for realizing the vision of self-driving networks. The automation of repetitive, well-defined, compute-intensive tasks reduces the load on human operators who currently "drive" the network and can be expected to gradually lead to the development of self-driving networks that are capable of sensing their environment and operating with little or no human assistance. To simplify the discussion and avoid confusion, our focus here is on network automation, especially because network automation already allows us to elucidate many of the issues and arguments that are also relevant for self-driving networks.

This report summarizes the main topics that were discussed during the course of this workshop and also includes a list of workshop attendees and the workshop agenda.

3 Data Collection

Academic networking researchers have traditionally obtained much of their data in a piecemeal and ad-hoc fashion, often as an after-thought, with hardly any useful metadata and generally with little (if any) attention given to data-cleaning issues. As a result, the network data that is typically available to them for the purpose of automating network management or operations tasks is rare, not necessarily representative, often a by-product of some other measurement activities, and typically of only very limited use for the task at hand. Even worse, labelled data that is key to applying some of the existing AI/ML techniques to network-specific problems is largely non-existent. Even networking researchers in industry struggle to collect data from their production networks in a flexible or scalable manner. For example, large mobile providers collect performance data from their end users' mobile phones but typically lack a comprehensive monitoring infrastructure that is capable of providing a holistic view of their network traffic across the different layers of the protocol stack, continuous in time, and with network-wide visibility. Similarly, some of the large content providers leverage tools that actively measure latency from hundreds of millions of their clients to a number of their globally distributed servers. They then use this information for automating tasks that ensure that the providers' infrastructures can attain their full potential performance. However, those very same tools typically do not support the collection of passive measurements that would provide critical information for automating an alternative set of tasks.

In short, networking researchers lack easy access to open-source collections of curated, cleaned, and labeled datasets that fuelled the research in other areas, such as computer vision, self-driving cars, etc. More concretely, availability of IMAGENET [7], an extensive database of hand-annotated images, catalyzed usage of AI/ML in computer vision research in general and in visual object recognition in particular [7]. Similarly, availability of datasets such as ARGOVERSE [24] or NUSCENES [25] that contain, among other measurements, hundreds of thousands of LIDAR[1] sweeps ensured that researchers could avoid driving their poorly-instrumented vehicles for data collection. In contrast, today's networking researchers still have to do the "driving" themselves to collect whatever data they can obtain with their existing tools.

3.1 University/Campus Networks as New Data Source

To address the long-standing data problem that is widely recognized as being one of the biggest current roadblocks to network automation, a radical solution would be for academic researchers to join efforts with the engineers and operators in their university's IT organization and discuss terms and conditions for access to and use of the university or campus network as a real-world production network. When instrumented with the latest network monitoring and data collection solutions (*e.g.*, commercial products such as [30]), such real-world production networks could provide the type of data that can be viewed as the networking analogue of the IMAGENET database and that has the potential of fueling the use of AI/ML in the networking area.

In addition to performing network-wide, continuous, lossless, full packet capture at scale, when deployed in a real-world production network, these latest network monitoring and data collection solutions typically populate (distributed) data warehouses with every packet that enters or leaves the enterprise and index them together with every conceivable on-the-fly generated metadata to support fast search capabilities. By providing a single platform for storing, mining, and visualizing all collected network data, these data warehouses are the go-to places for information that may be requested by security-related, network or application performance-specific, compliance-related or other tasks, may require off-line or real-time processing, or may be used for performing back-in-time analysis.

While the existence of and access to such rich data warehouses will be critical for academic networking researchers to develop, train, implement, and evaluate their "learning" algorithms in support of network automation, their presence will also pose new challenges for both the owner of this data (*i.e.*, the university's IT organization) and the users of this data (*e.g.*, researchers, IT engineers). Addressing these challenges will require striking a delicate balance between flexibility, scalability, and privacy, and we discuss some of the outstanding issues next.

3.2 Enabling Explorations: From Manual to Automated Feature Engineering

Traditionally, network researchers trying to leverage AI/ML methods for their work have been preoccupied with feature engineering—using domain knowledge of specify the features that ultimately make the chosen AI/ML-based method work (see for example [6]). However, as the complexity of the network automation tasks at hand increases and the number of possible features grows exponentially with the capabilities of modern network monitoring solutions, domain knowledge is in general no longer sufficient to decide on the set of most relevant features for a given problem. One solution to this challenge is to enhance existing AI/ML methods so they are capable of efficiently combing through the enormous feature space on their own to identify patterns or combinations of features that are best suited for solving a particular problem such as automatically inferring a specific network event.

On the one hand, recent developments in deep learning for automated processing of images, text, and signals have shown that for those types of data, it is possible to achieve significant automation in feature engineering (*e.g.*, see [17] and references therein). At the same time, feature engineering for network data has remained an intuition-driven, iterative, manual, and thus time-consuming activity. Therefore, it is bound to become a crucial bottleneck, preventing network researchers from reaping the full benefits and efficacy of leveraging AI/ML methods in their future work. However, the data warehouses populated by modern network monitoring and data collection solutions afford a unique opportunity for network researchers. These data warehouses enable automation of feature engineering for network data as part of their current exploratory efforts of developing suitable learning models for a range of different tasks related

[1] LIDAR (short for light detection and ranging) are the sensors responsible for generating the voluminous and high-quality measurements that have revolutionized the use of AI/ML applications for self-driving cars.

to network automation. In fact, for these efforts that will occupy network researchers for the foreseeable future, the superb quality and large quantities of all different kinds of network data available in these data warehouses will be of critical importance. Availability of high-quality data will aid the development of various network automation tools with enough built-in "intelligence" to benefit and ultimately replace a human performing these tasks today.

3.3 Enabling Flexibility: From Collection-Driven Data Analysis to Analysis-Driven Data Collection

Data collection efforts aimed at supporting exploratory efforts are most beneficial when they are by and large unconstrained in terms of what data should be collected, how it should be collected, and from where it should be collected. In this sense, the data warehouses populated by modern network monitoring and data collection solutions provide in principle all the flexibility a researcher can wish for in their pursuit of developing a tool for a specific network automation task that fully mines the available information and leverages automated feature engineering as part of learning a suitable model for the task at hand. Clearly, to achieve this goal, the AI/ML pipeline is an off-line process that is similarly unconstrained in terms of available compute resources, available time, and access to available data.

In contrast, assuming such an off-line learning model for a given network automation task is made available, its use in a real-world production network requires that it can operate at line rate. Aside from the challenge of running a potentially complex learning model at line rate (see our discussion in Section 5), there is also the need to sense or monitor the features required by the model in real time.[2] However, traditional data collection depends heavily on the choice of collection tool or protocol, and most of these tools or protocols offer very limited flexibility. In fact, the data collection decisions are often hard-wired or difficult to modify, typically resulting in the collection of data that is either too sparse or too voluminous. Importantly, unlike conventional network-management practice that embraces *collection-driven data analysis*, network automation requires *analysis-driven data collection*. That is, by letting the analysis determine what data to monitor or collect, it becomes feasible to only record the data for the features that are pertinent for the learning model or task at hand, even if the definition of what is pertinent changes over time. Such an approach will require developing a programming abstraction that makes it easier to express what, how, and from where to collect data in a programmable fashion at line rate. Such a programming abstraction also needs to simplify data collection across a diverse set of targets (PISA switches, RasPIs devices, etc.), protocols (ICMP, NetFlow, etc.), and infrastructures (RIPE Atlas, speedchecker, etc.).

3.4 Enabling Scalability

We have seen that continuously collecting and storing all the data from a network has enormous benefits when it comes to supporting network researchers in their efforts to explore the off-line development of suitable learning models for various network automation tasks. However, for any given set of tasks, not all collected data is of relevance, and being too aggressive in terms of data collection can be wasteful, especially when data collection is confined to the data plane and is required to be done in real time. To address this challenge, network researchers have begun to leverage programmable data planes to enable event-triggered or reactive data collection. For example, instead of collecting all packets in a complex network, a network operator at Alibaba reported on a system for debugging faults that only collects event-specific packets in their data center networks.

To date, reactive data collection solutions have been largely ad-hoc in nature and generalizing them so they are applicable to a broad range of different situations has been proven difficult. To scale reactive data collection, new architectural designs are needed that support task-specific and autonomous decision making on when to collect what data. A promising initial solution to this problem is Sonata [12], a query-driven streaming network telemetry system that leverages the hierarchical nature of IP addresses to scale data collection (and analysis). However, since hierarchical structures in network data are not limited to IP addresses, efforts on network automation can further benefit from algorithms that are able to more fully leverage the hierarchical structure inherent in network data.

Moreover, any given set of network automation tasks will require the extraction of multiple features from the network traffic. However, packet-level processing for features for different tasks can be similar. To fully exploit possible synergies and avoid redundant packet processing, it will be important to design data

[2] Here, we do not consider the option that the feature-specific data streams could be requested from the deployed network monitoring and data collection solution but assume instead an appropriately-instrumented data plane.

collection algorithms that are capable of fully leveraging the similarities that may exist between different features so as to optimize the use of the available but scarce network resources.

3.5 Preserving Privacy

On the one hand, the ability to collect, store, and inspect pretty much any aspect of traffic in a real-world production network creates exciting new opportunities. Researchers are afforded unique opportunities to make significant contributions to the area of network automation; network operators and IT engineers are provided with unprecedented visibility into their network. At the same time, this ability also creates new sets of challenges for the network operator or IT organization because, as the owners of the collected data, they are responsible for safeguarding it, including protecting user privacy. In particular, as data owner, issues such as what sort of data can/should or cannot/should not be collected and/or stored or what data can/should (not) be safely shared with third parties without violating user privacy or compromising an IT organization's policies are of paramount importance. Today, there is a significant difference between the legal policies for data privacy and the legal policies for data collection. For example, in the context of network data, network operators typically struggle with the definition of personally identifiable information (PII) attributes and as a result, most of them prefer and opt for a policy of minimal data collection and sharing.

To address these challenges that arise from the proposed permissive mode of network monitoring and data collection, we need to design data models that are not only amenable to legal reasoning but also make it easy to express data collection and sharing policies. Packets in a network carry a significant amount of information, some of which can be used to personally identify users and/or their online behavior. However, for many network management tasks, such sensitive information is not needed and could be anonymized, possibly in some structure-preserving (*e.g.*, prefix-preserving) manner, depending on the task at hand. Today, such anonymization is typically performed offline and may require significant efforts and/or resources. However, with the emergence of programmable data planes, knowing which network traffic features need to be extracted for a given task and which of them also have to be anonymized in ways that satisfy the data owner's privacy policies and do not interfere with the data user's objective of successfully automating the desired network management task suggests an attractive alternative approach. In particular, it argues for designing a system that leverages modern programmable switches (*e.g.*, commercial products such as [29]) to flexibly anonymize arriving packets at line rate (*i.e.,* in the data plane itself) [18]. Moreover, with such a system, the data owners have now the option to only store the packets in the data warehouse with their anonymized information in case they deem the original information to be too sensitive.

An alternative approach to addressing the privacy problem surrounding network data is inspired by recent developments in the area of computer vision research, where generative adversarial networks (GAN) have been used with great success to generate photo-realistic images of faces (*e.g.*, deepfakes), to age face photographs, or to produce videos of a person speaking given only a single photo of that person. Unfortunately, network data is in many ways very different from images or photographs, but recent work presented by one of the workshop participants has shown that GANs can be leveraged to synthesize simple network data in the form of time series while achieving good fidelity, flexibility, and privacy. However, given the complex nature of real-world network traffic data, more work is required before the use of synthesized packet-level network data with privacy-preserving sproperties for a wide range of features can be assessed in practice.

In general, instead of viewing data privacy as a possible roadblock for advancing the state-of-the-art in network automation research, we argue that it should be considered as an opportunity. Clearly, past experiences with and highly-publicized breaches of data privacy in the private sector have revealed rather questionable attitudes towards data privacy in an industry where monetizing network data (*e.g.*, who is doing what) is the primary goal. However, in the context of research on network automation, where the primary goal is to use network data for improving network management (*e.g.*, performance, security), we argue that there a great opportunity for revisiting the data privacy problem and demonstrating that data privacy can go hand-in-hand with network automation-related efforts without impeding on the data owner's responsibilities and the data user's objectives.

4 Data Analysis

Most data-analytic tasks for network automation require searching for patterns to detect various types of network events, be they performance or security-related. In recent years, most data analysis efforts in the networking area have evolved from using traditional statistical or conventional machine learning techniques to applying more complex deep learning methods. In theory, network operators can apply these tools to search for patterns in the data at hand to detect the onset of network attacks, localize performance bottlenecks, predict future network behavior, or infer other types of network events of interest. However, even though network operators have been applying state-of-the-art AI/ML tools to solve some specific problems in their network, in practice, their ability to use these tools more broadly, consistently, and effectively is often limited by (i) the quality of the available data (*e.g.*, noisy, sparse, encrypted, not time-synced), (ii) a general lack of labelled data, and (iii) the distributed nature of the collected data.

4.1 Dealing with Encrypted Traffic

With more and more of today's network traffic being encrypted, the inability to analyze this type of collected traffic beyond the transport layer makes it more difficult for network operators to detect patterns of interest. This proliferation of encryption is especially problematic for network operators of the large ISPs (*e.g.*, Comcast, Verizon, AT&T, etc.) whose end users expect a certain level of Quality-of-Experience (QoE) for the different (encrypted) applications or services that they are running.

Recent efforts (*e.g.*, NetMicroscope [26], Requet [14]) have shown that it is possible to develop learning models that can detect various network events without requiring deep packet inspection. For example, Requet [14] demonstrates that it is possible to predict different QoE metrics (*e.g.*, resolution, state, etc.) for encrypted video traffic in real time. With this information, network operators at ISPs can, for example, automate packet scheduling for the cable modem termination system (CMTS) that handles video traffic delivery for an ISP's downstream clients.

At the same time, for security-related tasks in particular, some traffic has to undergo deep packet inspection. However, in practice, the number of packets that require such in-depth inspection is typically only a small fraction of the total number of packets. This observation can be leveraged to selectively or/and on-demand decrypt network traffic for deep packet inspection. One way to to achieve this goal is to take advantage of recent advances in the area of network authorization (*e.g.*, FLANC [11]) that are concerned with automating the decision making for reactive deep packet inspection. However, more work will be required to build full-fledged system that integrate the different components that are necessary for scaling the data analysis for encrypted network traffic.

4.2 Labeling Network Data

While the development of reactive and flexible network monitoring and data collection systems or infrastructures can address many of the problems caused by the quality of the collected data (*e.g.*, noise, sparseness, encryption), generating appropriate labels for network data is a well-known hard problem. Labelled data is a pre-requisite for training the class of supervised machine learning algorithms, but network operators who would be best at assisting in or performing data labeling are typically not available because they are busy managing their network. Other areas such as computer vision has relied on crowd-sourcing for data labeling, but given the level of domain expertise required in the networking area, scaling crowd-sourcing solutions for labeling network data is challenging if not infeasible.

One option to address this problem and minimize the data labeling workload on network operators is to design learning algorithms that can leverage recent advances in the area of active learning [1, 27]. Active learning algorithms iteratively identify the data for labeling, reducing the total number of data points that require manual labeling for a learning problem. In a similar fashion, recent works in the area of representation learning also show how to build learning models that can make the best use of limited labels [5]. Alternatively, to avoid the problem with labeled data altogether, networking researchers need to more fully explore the use of unsupervised machine learning algorithms for analyzing network data.

4.3 Distributed Streaming Data

Searching for patterns in network data is significantly more difficult when the data is distributed across different nodes in the network, without a central location where all the data is stored in a single place and can be analyzed "in place." Even for off-line analysis, the collected network data at each node is in general too voluminous to send to a central storage location and requires that the analysis of such distributed data itself be distributed. Even more demanding is the case where the real-time constraint imposed on the tools designed to perform specific network automation tasks requires network data to be treated as streaming data that is in addition often distributed in nature. The analysis of distributed streaming data typically assumes that while there exists a central node or controller that can communicate and coordinate with each remote node, the objective is to develop analysis methods that have minimal communication overhead (*e.g.*, to satisfy the real-time constraint and not overwhelm the communication channels). It is the controller's job to interact with the remote nodes in such a way that it receives the information it needs to perform the inference task at hand.

A textbook example of such a task is the detection of network-wide heavy hitters; that is, flows that exceed a certain threshold for a given metric of interest (*e.g.*, number of bytes, number of distinct source IPs). Being able to identifying heavy-hitters in an online and network-wide fashion is critical for tools that are designed for, say, detecting and mitigating certain DDoS attacks on an enterprise network as they happen (*i.e.*, in real time). Recent efforts show how to leverage distributed streaming data and modern-day programmable switches to design a distributed heavy-hitter detection scheme for identifying network-wide heavy-hitters in real time [15].

While over the years, the CS community has developed a myriad of algorithms for answering a number of different questions about single streaming data and even distributed streaming data (*e.g.*, heavy-hitter detection), much work remains with respect to developing network telemetry systems that can support a broad range of different network automation tasks. In view of recent efforts such as [12], to perform query-driven and network-wide telemetry at scale, future network telemetry systems will have to carefully orchestrate the flexible processing capabilities of modern-day programmable devices and stream processors and fully exploit the synergies that arise from multiple tasks requesting similar input or resulting in similar data processing pipelines. In this context, recent advances in federated learning (*i.e.*, a distributed machine learning approach which enables model training on a large corpus of decentralized data [20]) and secure federated learning (*i.e.*, an approach that allows data and knowledge to be shared among different entities without compromising user privacy [31]) suggest opportunities for applying distributed machine learning methods to distributed streaming data while ensuring data privacy.

5 Closing the Loop

The vision of performing network automation tasks efficiently at scale and in real time requires executing a myriad of decision making processes over very short time scales. Such intense decision making is beyond human capabilities and results in human operators becoming the bottleneck. As today's large cloud, content, and Internet service providers are experiencing this "pain point" first hand, they are prioritizing the development of more automated tools that are designed to reduce the workload for human operators. As the complexity and scale of their networks increase, designing effective tools that assist human operators in their decision making will become more important and at the same time more difficult.

We consider efforts that "automate what can be automated" and that focus on improving the accuracy of the various task-specific learning models to minimize the decision space to be promising first steps towards realizing the idea behind network automation; that is, keep human operators in the loop but leverage their expertise for handling "surprises" (*e.g.*, unexpected behaviors that defy easy explanations) and not for dealing with the more mundane tasks associated with the day-to-day operations of typical production networks. As advances in network automation will gradually transform network operations and management into a less human-centric and more machine-driven activity, we envision a number of challenges that require the network researcher's attention.

5.1 Developing Unified Programming Abstraction

Recent works have focused on developing better programming abstractions for network monitoring [12,23] and control [13,16,19,21,22,28]. However, a programming abstraction that unifies network monitoring and control with autonomous decision making is still missing. The absence of such a unified programming abstraction negatively impacts the development of robust and reliable solutions for network automation tasks at scale. Such tasks often consist of a separate detection and mitigation module that have to work in tandem to accomplish the task in real time. In fact, in control-theoretic terminology, to perform detection and mitigation in real time means being able to "close the (control) loop"—sensing the network for the "right" data, performing the "right" automated inference, and applying the "right" programmatic control.

Developing such a unified programming abstraction that not only makes the best use of the available but limited network resources but can also leverage state-of-the-art AI/ML tools and libraries is challenging. While recent efforts such as Sonata [12], Marple [23], etc. that rely on analytics-friendly data flow abstraction are promising first attempts, more work will be required to develop the type of programming abstractions that are capable of unifying network monitoring, automated inference, and programmatic control. Moreover, such abstractions should be directly applicable to distributed settings.

5.2 Addressing the Trust Problem

If AI/ML-based tools are used at all in some of today's production networks, they typically entail simple learning models such as decision trees and SVMs. In fact, decision trees and their variants are currently one of the most commonly-encountered ML techniques in support for automated decision making in real-world production networks. They are in general preferred over the latest deep learning models, mainly because they are more lightweight, less complex, and more intuitive. While suitable for simple problems, their traditional use is usually ill-advised when the problem at hand is more complex. For example, using a decision tree model is not a recommended solution for a problem of practical interest for mobile providers—identifying rogue mobile devices using time series of various features collected from tens of millions of mobile devices.

However, one of the main reasons why today's network operators favor simpler learning models over more sophisticated and complex ones is that the latter are in general used as "black boxes", unable to provide any insights or understanding. A keen desire for wanting to know when these black boxes succeed or fail (and why) and understanding why they produced a certain result and not something else is at the heart of the network operators' demands for cracking these black boxes open and turning them into "white boxes" so that as users, they are able to understand, explain, and interpret their output; that is, trust them.

Other areas where AI/ML is increasingly used and where failures can be catastrophic (*e.g.*, autonomous vehicles, medical diagnosis, and legal decision making, etc.) also faced similar trust issues. Recognizing that the effectiveness of these systems is limited by the inherent ability of most of these learning algorithms to explain decision and actions to human experts, in 2017 DARPA launched a new program on "Explainable Artificial Intelligence (XAI)" [10]. The stated aims of XAI are *"to create a suite of AI/ML techniques that (i) produce more explainable models, while maintaining a high level of learning performance (prediction accuracy), and (ii) enable human users to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners."* XAI is further expected to contribute to the development of "third-wave AI systems', where learning algorithms *"understand the context and environment in which they operate, and over time build underlying explanatory models that allow them to characterize real world phenomena."*

Since the launch of the XAI program, we have witnessed exciting developments that hint at what may be possible in terms of explainable or interpretable AI/ML models. In particular, we can surmise why they promise to be a good match for the area of network automation where understanding how these advanced algorithm do what they do will be imperative for network operators/engineers and security analysts before they are willing to hand over consequential decision making to AI/ML-based tools and deploying them in their networks. To illustrate, a series of recent papers [2–4] describes new efforts for interpreting or explaining black-box models via model extraction. The basic idea is to approximate a complex black-box model using a more interpretable or explainable model. In particular, by considering decision trees as approximate models and assuming that the approximation quality or "fidelity" is high, any issues in the complex black-box model should be reflected in the approximate model (*i.e.*, the extracted decision tree). Since decision trees are highly interpretable, users can now start explaining how the corresponding black-box does what it does by examining the extracted high-fidelity decision tree instead.

The feasibility of the idea behind explainable or interpretable AI/ML, including algorithms for extracting high-fidelity decision trees, has already been demonstrated in the context of black-box models such as random forests, DNNs, and some classical reinforcement learning models. In these documented cases, the constructed decision trees have been successfully used to debug and interpret the considered supervised learning models and to understand the control policies learned from traditional reinforcement learning models. However, despite these promising advances, there are many open problems and much remains to be done in this ongoing effort to transform AI/ML's black-box models into "white boxes." For example, decision trees quickly lose their interpretability or explanatory power as their depth increases and the number of paths through the tree becomes unwieldy. It will be important to explore alternative approximate models that match the high interpretability of decision trees and can be proven to also have high fidelity across a range of commonly-used black-box models. By being able to address these and similar issues (see for example [8]), it may be possible to persuade networking researchers to embrace explainable or interpretable AI/ML in their quest for gaining the trust of the network operators who are in charge of deploying AI/ML-based tools in tomorrow's network in an attempt to gradually automate more and more of the network's day-to-day operations.

6 Road to Deployment

Viewing network automation where the network operator remains in the loop (*i.e.*, the operator still "drives" the network but many of the more mundane tasks no longer require the operator's attention) as a critical first step towards the development of autonomous networks that largely "run by themselves", it is important to understand the road-to-deployment for newly proposed network automation solutions. As network operators of some of the large content, cloud, and Internet service providers are announcing publicly their interest in and pursuit of AI/ML-based solutions for managing their increasingly complex infrastructures and services, they typically focus on problems that are dictated by their business requirements and practical needs. For example, upon indications of persistent congestion, cloud/content providers (*e.g.*, Google) want to quickly upgrade their networks' capacity with minimal human intervention. On the other hand, large ISPs with a large mobile user base (*e.g.*, Verizon) are interested in automating the detection of rogue mobile devices in real time, and broadband service providers (*e.g.*, Comcast) look towards automation when it comes to the detection, identification of root causes, and resolution of service-related complaints from their broadband customers.

However, how do these network operators go about "road-testing" their AI/ML-based solutions before deploying them in their production networks? Put differently, for an academic researcher who designed and developed some AI/ML-based solutions for a particular network automation task, what is the road to deployment. That is, what steps are necessary for the researcher to convince a network operator to hand over the task at hand to a tool that relies on some AI/ML model for the type of decision making that the human operator is intimately familiar with based on her domain knowledge and past experience – having performed this task herself routinely in the past? Some network operators mentioned that they rely on the vendors from whom they purchase some of these solutions, this answer begs the question of how these vendors go about road-testing this new generation of AI/ML-based solutions to the point where they are convinced of their commercial success in deployment.

6.1 Enabling Self-driving Networks with Network Automation

With network automation being the first step towards realizing the vision of self-driving networks, it is of great practical interest to understand the ingredients that matter the most for ensuring a viable and sensible road-to-deployment for newly proposed AI/ML-based network automation solutions. In this workshop, we broadly identified a five-step process for how in general a newly proposed network automation solution currently evolves or transitions from a research idea into a deployed tool in a production networks.

1. Network operators often resolve problems in an ad-hoc manner, piggybacking on data collected using standard tools such as ping, traceroute, or NetFlow.

2. For problems that: are more business-critical, impact the network operator's bottom line, and cannot be detected using existing tools; more specific data collection tools/systems are designed and developed to resolve those problems.

3. The datasets collected in the first or second step form the basis for developing learning algorithms that typically leverage traditional black-box models. This development step is in general performed offline so that the available data can be fully utilized, the training phase is not constrained by time or compute resources, and the learning algorithm's properties (*e.g.*, performance, accuracy, robustness) can be evaluated in simulations and/or emulations.

4. The learning algorithms that result from the third step are used in testbed settings or in real-world production networks (read-only mode, though) to iterate and improve the algorithms' end-to-end workflow.

5. After successful completion of step four, the learning algorithms are deemed sufficiently reliable, safe, robust and accurate to be deployed in a production setting incrementally (*e.g.*, gradually increase the portion of overall traffic that is handled by the deployed solution).

However, given the recent advances in the area of AI/ML, tomorrow's network operators will face a critical choice. By favoring the deployment of ever-more powerful AI/ML-based solutions in their production network, they are at the same time agreeing to hand over more and more consequential decision making to solutions that leverage ever-more complex black box models. We fully expect that when faced with this choice today or in the foreseeable future, these network operators will ask for or require further or alternative pieces of evidence that will convince them that these new solutions are "safe" and can be trusted. To this end, the third step in the outlined five-step process will require more efforts and new ideas, and we elaborate below on how this third step could be embellished.

6.2 University/Campus Networks as New Testbed

We already argued in Section 3 why academic researchers should start considering and leveraging their university/campus network as a real-world production network. In this earlier context, our suggested use case for these networks was that when properly instrumented with state-of-the art network monitoring and data collection solutions, they provide enormous value as a new source of network data of unprecedented quality and quantity. By leveraging their full potential as new data sources, university/campus networks are promising targets for finally getting access to the "right" data that will fuel the application of AI/ML in the networking area.

In particular, assuming that data privacy-related issues can be be adequately dealt with (see Section 3.5), with access to the "right" data, academic network researchers will be in a unique position to carefully examine the relevance, suitability, and applicability of the idea behind XAI (see Section 5.2) for the networking area in general and the area of network automation in particular. Given the importance of the AI/ML trust issue for network operators, the expected proliferation of robust, scalable, and explainable learning models in the field of networking promises to provide a level of expertise and experience with regard to explainable or interpretable AI/ML models that may allow network researchers to gauge the full potential of XAI. Importantly, being able to demonstrate with concrete examples how to extract from complex and heavy-weight black boxes light-weight and interpretable learning models that provide network operators with new insights and understanding will go a long way in clearing their road to deployment.

In theory, with the "right" data and the "right" AI/ML in place, network researchers should be able to invigorate research efforts in the area of network automation. In practice, however, a key obstacle remains, namely how to "road-test" a tool that has been developed for a particular network automation task, typically includes a learning algorithm that has been developed and/or evaluated offline or in a sterile testbed environment, and has never been deployed in a actual production network? Just as in the case of self-driving cars where it is necessary but not sufficient to use test tracks, it would be ill-advised to deploy AI/ML-based network automation tools without any prior testing and training in real-world production settings. However, given how network disruptions can affect enterprises or entire network infrastructures, network operators are inherently risk averse and shy away from deploy untested solutions directly in production settings.

To overcome this obstacle, we suggest treating university/campus networks as real-world production networks. This time, when properly instrumented with state-of-the-art programmable devices, these networks can take on the role of various types of testbeds for road-testing AI/ML-based solutions.

On the one end of the spectrum, a proposed solution can be deployed directly in the production network itself, but initially strictly in *read-only* mode. This way, the testbed is the production network and provides an "emulation" with highest fidelity. To minimize the possibility of disrupting the production traffic, in a first step, only the data collection and data analysis actions are enabled, but no control actions are executed. By logging the predicted control actions and relying on human operators to assess the actions' correctness, the system can be made more robust and reliable over time, incrementally delegating the decision making from the human operator to the deployed AI/ML-based solution.

On the other side of the spectrum, network monitoring and data collection and analysis at scale may require specialized devices and protocols that might not be available in the production network in question. The unavailability of such devices can affect the exploration of new and forward-looking solutions for data collection and analysis. One way to deal with this problem is to build a dedicated testbed that is equipped with state-of-the-art hardware and software solutions (*e.g.*, for network monitoring and data collection and for programmability of the data plane). By mirroring the production network's traffic to such a testbed, it is possible to perform the all of the network data collection actions, the data analysis actions as well as apply control actions in the testbed. However, mirroring all of the production network's traffic entails high bandwidth overhead and may rule out the simple approach of sending all packets from the production network to the testbed. In this case, to lower the bandwidth overhead, new approaches are needed to selectively sample relevant traffic from the production network and mirror only that sampled traffic to the testbed. Such an approach is trading off fidelity against flexibility but has the added benefit that data privacy issues may be less a concern for the sampled traffic as compared to the production network's complete traffic.

7 Conclusion

By viewing network automation, where human operators still "drive" the network while automating mundane (repetitive) tasks, as a critical first step towards the development of self-driving networks, this workshop mainly concentrated on identifying new research opportunities for academic network researchers in the area of network automation. In particular, our focus was on how to best leverage technological advances in relevant fields (*e.g.*, network monitoring and data collection, AI/ML, programmable data planes) to advance the state-of-the-art in network automation.

In terms of actionable items, concrete suggestions include a call-to-action for academic researchers to consider their university/campus network as a real-world production network and start leveraging it for their research. To this end, we propose that such a university/campus network should be adequately instrumented with (i) the latest network monitoring and data collection solutions so it can serve as a unique source of the all-important network data, and (ii) state-of-the-art programmable devices so the network can also function as a premier testbed for road-testing newly-proposed AI/ML-based network automation solutions. While the technologies for such instrumentation at a university/campus network exist today and can be readily acquired at a reasonable cost, their actual deployment will rely on close collaborations with the university's IT organization that has the sole responsibility for managing and operating the university/campus network.

At the same time, we also identify new research challenges that concern related data privacy problems and arise from the current AI/ML trust problem that prevents today's network operators from deploying AI/ML-based black box solutions in their production network. Solving these critical problems will require a concentrated effort by the networking research community and close collaboration with data privacy experts and the AI/ML research community. Importantly, by proposing to fully exploit the opportunities afforded by real-world production networks such as university/campus networks, we outline an agenda for network automation research that promises to overcome the long-standing problems of not having access to the "right" data and not being able to evaluate proposed solutions in fully operational settings.

References

[1] P. Bachman, A. Sordoni, and A. Trischler. Learning algorithms for active learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR, 2017.

[2] O. Bastani, C. Kim, and H. Bastani. Interpretability via model extraction. In *Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2017.

[3] O. Bastani, C. Kim, and H. Bastani. Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504*, 2017.

[4] O. Bastani, Y. Pu, and A. Solar-Lezama. Verifiable reinforcement learning via policy extraction. In *Conference on Neural Information Processing Systems*, 2018.

[5] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 2013.

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

[7] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[8] A. Dethise, M. Canini, and S. Kandula. Cracking open the black box: What observations can tell us about reinforcement learning agents. In *ACM SIGCOMM NetAI Workshop (to appear)*, 2019.

[9] N. Feamster and J. Rexford. Workshop on self-driving networks. 2018.

[10] D. Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA)*, 2, 2017.

[11] A. Gupta, N. Feamster, and L. Vanbever. Authorizing Network Control at Software Defined Internet Exchange Points. In *ACM Symposium on SDN Research (SOSR)*, 2016.

[12] A. Gupta, R. Harrison, A. Pawar, M. Canini, N. Feamster, J. Rexford, and W. Willinger. Sonata: Query-Driven Network Telemetry. In *ACM SIGCOMM*, 2018.

[13] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett. SDX: A Software Defined Internet Exchange. In *ACM SIGCOMM*, 2014.

[14] C. Gutterman, K. Guo, S. Arora, X. Wang, L. Wu, E. Katz-Bassett, and G. Zussman. Requet: real-time qoe detection for encrypted youtube traffic. In *ACM Multimedia Systems Conference*, 2019.

[15] R. Harrison, Q. Cai, A. Gupta, and J. Rexford. Network-wide heavy hitter detection with commodity switches. In *Symposium on SDN Research*. ACM, 2018.

[16] X. Jin, J. Gossels, J. Rexford, and D. Walker. Covisor: A compositional hypervisor for software-defined networks. In *USENIX NSDI*, 2015.

[17] J. M. Kanter and K. Veeramachaneni. Deep feature synthesis: Towards automating data science endeavors. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2015.

[18] H. Kim and A. Gupta. Ontas: Flexible and scalable online network traffic anonymization system. In *ACM SIGCOMM NetAI Workshop (to appear)*, 2019.

[19] H. Kim, J. Reich, A. Gupta, M. Shahbaz, N. Feamster, and R. Clark. Kinetic: Verifiable Dynamic Network Control. In *USENIX NSDI*, 2015.

[20] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[21] J. C. Mogul, A. AuYoung, S. Banerjee, L. Popa, J. Lee, J. Mudigonda, P. Sharma, and Y. Turner. Corybantic: towards the modular composition of sdn control programs. In *ACM HotNets*, 2013.

[22] C. Monsanto, J. Reich, N. Foster, J. Rexford, and D. Walker. Composing software defined networks. In *USENIX NSDI*, 2013.

[23] S. Narayana, A. Sivaraman, V. Nathan, P. Goyal, V. Arun, M. Alizadeh, V. Jeyakumar, and C. Kim. Language-directed hardware design for network performance monitoring. In *ACM SIGCOMM*, 2017.

[24] Argoverse. `https://www.argoverse.org/data.html`.

[25] Nuscenes. `https://www.nuscenes.org/overview`.

[26] P. Schmitt, F. Bronzino, R. Teixeira, T. Chattopadhyay, and N. Feamster. Enhancing transparency: Internet video quality inference from network traffic. *TPRC 46: The 46th Research Conference on Communication, Information and Internet Policys*, 2018.

[27] B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[28] P. Sun, R. Mahajan, J. Rexford, L. Yuan, M. Zhang, and A. Arefin. A network-state management service. In *ACM SIGCOMM*, 2014.

[29] Barefoot's Tofino. `https://www.barefootnetworks.com/technology/`.

[30] Niksun's NetVCR. `https://www.niksun.com/c/1/ds/NIKSUNDatasheet_NetVCR.pdf`.

[31] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2), 2019.

A Appendix: Schedule and Participants

A.1 Schedule

For breakouts, we asked the workshop participants to form two groups, one focusing on enabling self-driving networks for eyeball networks, and the other for cloud/content provider networks. More specifically, the first group focused on issues related to managing campus, home, broadband, enterprise, and mobile networks; and the second group covered topics related to managing inter/intra-data center, and wide-area networks.

Table 1: *Workshop schedule*

Thursday, April 4, 2019	Venue: Palmer Room, Nassau Inn
10:00–10:30 am	Breakfast
10:30–11:00 am	Introduction/Setting the Stage (Arpit Gupta, Walter Willinger)
11:00–11:45 am	**Session 1: Data Collection** Two talks from cloud/content providers on their perspective on where and how self-driving networks can help and on what challenges did they faced building a data collection infrastructure at scale. • Mukarram (Google): Self Driving Network Systems Management • Matt Calder (Microsoft): Internet Measurements for Self-driving Networks
11:45 am–1:15 pm	**Breakout Session 1: Data Collection** How to collect diverse and high-quality network data that can be used for training and evaluating new learning models specifically designed to enable self-driving networks?
1:15–2:15 pm	Lunch (on your own)
2:15–3:00 pm	**Session 2: Data Analysis** Two talks related to problems arising in the context of data analysis (e.g., analyzing encrypted network data, building a synthetic dataset using GAN). • Paul Schmitt (Princeton University): Internet Video Quality Inference from Encrypted Network Traffic • Giulia Fanti (CMU): Building High-Fidelity Synthetic Datasets using Generative Adversarial Networks
3:00–3:15 pm	Coffee Break
3:15–4:15 pm	**Breakout Session 2: Data Analysis** How to process (online/offline) sparse, noisy, encrypted network data to concurrently infer various network events at scale?
4:15–5:30 pm	**Readout from Sessions 1 and 2**
5:30–6:30 pm	Reception at Palmer House, Princeton University, 1 Bayard Lane, Princeton, NJ
6:30 pm onwards	Dinner (same place)
Friday, April 5, 2019	
8:30–9:00 am	Breakfast

Table 1: *Workshop schedule*

9:00–10:00 am	**Session 3: Closing the loop** Three talks on building systems for enabling aspects of self-driving networks in different settings. • Bryan Larish (Verizon): AI Innovations in Service Provider Networks • Gilberto Mayor (Quaasar Inc.): An ML Platform for Managing & Improving QoE for Broadband & WiFi Networks • Harry Liu (Alibaba): In-network Telemetry in Alibaba's Datacenters
10:00–10:45 am	**Discussion: Closing the loop (no breakout)** How to design in-network measurement architectures that can adapt learning models and assess the performance of various reactive control actions in real time?
10:45–11: 15 am	**Readout from Session 3**
11:15–11:30 am	Coffee break
11:30–11:45 am	**Session 4: Testbeds for self-driving networks** One talk on providing academic researchers with real-world environments for pursuing research on self-driving networks. • Jack Brassil (Princeton): Campus Networks as Research Testbeds
11:45 am–12:45 pm	**Discussion: Testbeds for self-driving networks (no breakout)** How to develop testbeds and platforms that can be used to develop and deploy self-driving networks in realistic settings with minimal disruptions of normal network operations?
12:45–1:45 pm	Wrap-up and writing assignments

A.2 Participants

Academia (17):

- Rachit Agarwal, Cornell University
- Ran Ben Basat, Harvard University
- Jack Brassil, Princeton University
- Vladimir Braverman, Johns Hopkins University
- Giulia Fanti, CMU
- Nick Feamster, Princeton University
- Arpit Gupta, UC Santa Barbara
- Trinabh Gupta, UC Santa Barbara
- Xin Jin, Johns Hopkins University
- Ethan Katz-Bassett, Columbia University
- Joon Kim, Princeton University
- Shir Landau Feibish, Princeton University

- Vincent Liu, University of Pennsylvania
- Ratul Mahajan, University of Washington/Intentionet
- Srinivas Narayana, Rutgers University
- Jennifer Rexford, Princeton University
- Paul Schmitt, Princeton University

Industry (8):

- Francesco Bronzino, Nokia Bell Labs
- Matt Calder, Microsoft
- Ken Duell, AT&T
- Harry Liu, Alibaba Group
- Bryan Larish, Verizon
- Gilberto Mayor, Quaasar Inc.
- Mukarram Tariq, Google
- Walter Willinger, NIKSUN, Inc.

NSF (2):

- Darleen Fisher
- Ann von Lehman

Students (3):

- Xiaoqi Chen, Princeton University
- Zinan Lin, Carnegie Mellon University
- Robert MacDavid, Princeton University