Machine Learning for Network Control and Digital Network Twinning

Patrick Krämer
Technical University of Munich
patrick.kraemer@tum.de

Andreas Blenk
Technical University of Munich
andreas.blenk@tum.de

***Abstract*—This paper summarizes the most important work conducted in the field of machine learning and artificial intelligence applied to communication networks from the Chair of Communication Networks (LKN), Technical University of Munich. Furthermore, it gives an overview of current ongoing work and research projects, which target the vision of digital network twinning for autonomous network control.**

***Index Terms*—network control, network algorithms, autonomous networking, data-driven, digital twin**

I. Vision: Towards Digital Network Twinning

The vision of LKN at TUM is a fully data-driven approach to network operation and control. In this vision, network control algorithms are synthesized from network data (e.g., monitoring data, algorithm data), and tailored to the properties of specific networks. Fig. 1 illustrates LKN's vision. The basis forms the high quality data of real systems that is obtained, e.g., through measurements. The measurements enable the creation of Digital Twins (DTs) of the network systems. The DTs serve as unified interfaces to the networks, represent the networks' state, and allow what-if-analysis, i.e., show how networks could behave for specific control decisions. The Algo-College uses the DT's capabilities to evaluate the effect of control decisions, enabling the training of control algorithms that are tailored towards the network.

Our vision is enabled through the thorough application of Machine Learning (ML) at every stage: Network Measuring, Network Modeling, and Control Algorithm Training. Sec. II gives an overview of the chair's work. Sec. III informs about our current projects.

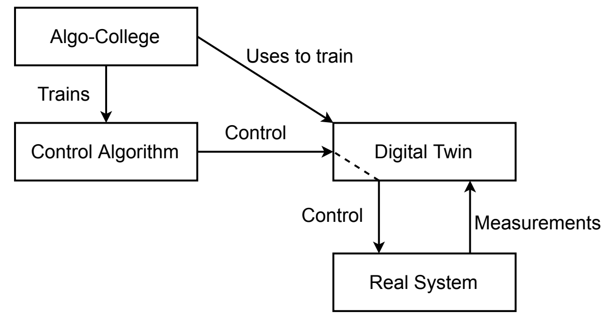


Fig. 1: Continuous system monitoring enables a Digital Twin of the system that ML uses to train control algorithms that control the system through the DT.

II. Past until Present

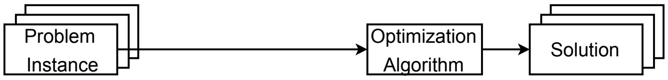
In the past, the chair focused on the data-driven design of network control algorithms, and methods to generate high-quality data for effective DT creation.

A. Data-Driven Network Optimization

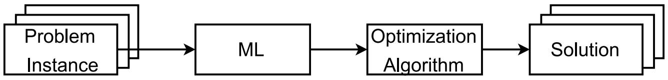
At the heart of network control often lies the solving of combinatorial optimization problems. Fig. 2 illustrates three ways towards solving such problems. Fig. 2a shows that traditional optimization algorithm, e.g., heuristic or Integer Linear Program (ILP), receives problem instances and solves them. This neglects the wealth of data that is generated in the process, i.e., the problem instances and solution pairs.

Fig. 2b shows a straightforward way to use the problem/solution pairs. ML uses the pairs to improve traditional optimization algorithms. In Blenk et al. [1], and ISMAEL [2] we use ML to learn the rejection behavior of compute intensive optimization algorithms. Here, ML serves as a cheap filter to avoid compute intensive executions of the algorithm on infeasible requests. In He et al. [3], and *o'zapft is* [4], we use ML to predict properties of optimization algorithms' solutions, such as the cost of virtual network embeddings, and if a node is likely to host an SDN controller. The predictions enable admission control, or a reduction of the solution space. While we show performance improvements with this approach, it is fundamentally limited by the performance and the quality of the actual optimization algorithm. However, even without explicitly learning from data, LKN demonstrates how NeuroViNE [5] uses Hopfield Networks to prune the solution space for Virtual Network Embedding (VNE) problems. As a result, NeuroViNE reduces the run-time of subsequent optimization algorithms, and improves its performance through eliminating bad local optima.

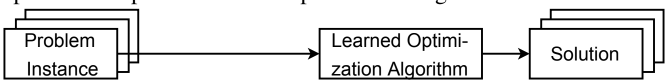
To overcome the traditional algorithms' limitations, Fig. 2c uses a learned algorithm instead. In addition to the problem/solution pairs, the learned algorithm takes the network's feedback into account, allowing ML to tailor algorithms to the properties and requirements of a specific network. Currently, we rely on simple abstractions of the network's behavior to achieve this. For example, AHAB [6] uses a high-level data-center network abstraction to learn admission control for distributed jobs. AHAB uses the model to simulate future developments and then decides whether to accept a job in the current state. Kalmbach et al. [7], uses an abstract routing



(a) Optimization algorithms solve problems and do not use the generated data.



(b) ML uses the generated data to provide information that helps to optimize the performance of optimization algorithms.



(c) ML learns an optimization algorithm from data and replaces traditional algorithms.

Fig. 2: Traditional and ML augmented network control.

model to learn network topologies that provide high path-diversity for one specific request pattern. ComNav [8], uses a packet-level simulation and Deep Reinforcement Learning (DRL) to learn a Neural Network-based protocol that incorporates network specific properties automatically. Kalmbach et al. [9] uses an Evolutionary Algorithm to optimize large placement problems using a simple model for edge-cloud infrastructure. `sfc2cpu` [10] optimizes the allocation of Virtual Network Functions (VNFs) to CPU cores with DRL. `sfc2cpu` improves the performance by learning how to avoid interference between VNFs from a simple model of the compute platform. Yet, our previous works in this area share the reliance on high-level, often hand-crafted models or simulations. In future work, a direct connection between real systems and DTs (or abstractions) should be established.

B. Adversarial Benchmarking for Networks and Network Representations

Our existing work often relies on abstract models that capture a network's high-level behavior and are oblivious to low-level effects that can have a strong performance impact. To learn control algorithms that are aware of low-level effects, we envision DTs that accurately reflect them. To get a DT that reflects a large range of the system's behavior, observing the system during operation might not be enough. Instead, deliberate measurements that trigger a wide range of system behaviors are necessary. To automate this process, TOXIN [11] and Netboa [12] show that it is possible to use AI/ML to obtain configurations that result in new experience. This avoids measurements with similar results, and can reduce the effort to obtain the necessary data for high-quality system models. To guide the automated design of experiments, the representation of a network state is important to capture the changes that a specific configuration triggers. Here, we used ML to automatically extract and visualize the behavior of a network from packet-level traces [13]–[15].

III. Ongoing work

We work on the different aspects of our vision, and strive towards integrating them in first small use-cases. Our research is supported through three projects. The goal of the Adversarial Design Framework for Self-Driving Networks (ADVISE) project is to investigate how predictable networks behave, design data-driven algorithms, and benchmark these algorithms with ML in an automated fashion. In the AI-NET Protect project, we are working towards the comprehensive monitoring, representation, and visualization of communication networks. In an industry cooperation with Rakuten Mobile we work on methods to obtain DTs from measurement data to use them to improve network controllers. We work towards culminating the projects into a system that allows a network to continuously improve itself in an automated fashion.

Acknowledgements

This work is partially funded by the Federal Ministry of Education and Research in Germany (BMBF) as part of the project AI-NET-PROTECT (grant ID 16KIS1294), and by the German Research Foundation (DFG) as part of the project ADVISE (grant ID 438892507).

References

[1] A. Blenk, P. Kalmbach, P. van der Smagt, and W. Kellerer, "Boost Online Virtual Network Embedding: Using Neural Networks for Admission Control," in *CNSM*, Oct. 2016.
[2] J. Zerwas, P. Kalmbach, S. Schmid, and A. Blenk, "Ismael: Using Machine Learning To Predict Acceptance of Virtual Clusters in Data Centers," *IEEE TNSM*, 2019.
[3] M. He, P. Kalmbach, A. Blenk, S. Schmid, and W. Kellerer, "Algorithm-Data Driven Optimization of Adaptive Communication Networks," in *IEEE ICNP*, Toronto, ON, Canada, Oct. 2017.
[4] A. Blenk, P. Kalmbach, S. Schmid, and W. Kellerer, "o'zapft is: Tap Your Network Algorithm's Big Data!" in *Big-DAMA*, Los Angeles, CA, USA, Aug. 2017.
[5] A. Blenk, P. Kalmbach, J. Zerwas, M. Jarschel, S. Schmid, and W. Kellerer, "NeuroViNE: A Neural Preprocessor for Your Virtual Network Embedding Algorithm," in *INFOCOM*, Apr. 2018.
[6] J. Zerwas, P. Kalmbach, C. Fuerst, A. Ludwig, A. Blenk, W. Kellerer, and S. Schmid, "AHAB: Data-Driven Virtual Cluster Hunting," in *IFIP Networking 2018*, IFIP, Ed., Zurich, Switzerland, May 2018.
[7] P. Kalmbach, J. Zerwas, P. Babarczi, A. Blenk, W. Kellerer, and S. Schmid, "Empowering Self-Driving Networks," in *SelfDN 2018*, ACM, Ed. ACM, 2018.
[8] P. Krämer and A. Blenk, "Navigating Communication Networks with Deep Reinforcement Learning," in *NetSys 2021*, vol. 80. Lübeck, Germany: ECEASST, Sep. 2021, p. 16.
[9] P. Kalmbach, A. Blenk, W. Kellerer, R. Pries, M. Jarschel, and M. Hoffmann, "GPU Accelerated Planning and Placement of Edge Clouds," in *NetSys*, Garching b. München, 2019.
[10] P. Krämer, P. Diederich, C. Krämer, R. Pries, W. Kellerer, and A. Blenk, "sfc2cpu: Operating a Service Function Chain Platform with Neural Combinatorial Optimization," in *IM 2021*, Bordeaux, France, 2021.
[11] S. Lettner and A. Blenk, "Adversarial network algorithm benchmarking," in *CoNEXT '19 Companion*. Orlando, FL, USA: ACM, Dec 2019.
[12] J. Zerwas, P. Kalmbach, L. Henkel, G. Retvari, W. Kellerer, A. Blenk, and S. Schmid, "NetBOA: Self-Driving Network Benchmarking," in *NetAI '19*. Beijing, China: ACM SIGCOMM, 2019.
[13] P. Kalmbach, A. Blenk, M. Klügel, and W. Kellerer, "Generating Synthetic Internet- and IP-Topologies using the Stochastic-Block-Model," in *AnNet*, May 2017.
[14] P. Kalmbach, F. Lipp, D. Hock, W. Kellerer, and A. Blenk, "NOracle: Who is communicating with whom in my network?" in *SIGCOMM'19*, ACM, Ed., Beijing, Aug. 2019.
[15] P. Kalmbach, A. Blenk, S. Schmid, and W. Kellerer, "Themis: Data Driven Approach to Botnet Detection," in *INFOCOM*, Honolulu, Hawaii, 2018.