Can You Trust AI-assisted Network Automation? A DRL-based Approach to Mislead the Automation in SD-IPoEONs

Min Wang[1], Siqi Liu[1], Zuqing Zhu[1]

1. University of Science and Technology of China, Hefei, Anhui 230027, China, Email: zqzhu@ieee.org

Abstract: We study the vulnerability of artificial intelligence assisted network automation (AIaNA), and design a deep reinforcement learning (DRL) model to mislead the AIaNA in software-defined IP over elastic optical networks (SD-IPoEONs) through crafting/injecting adversarial traffic samples.

1. Introduction

Recently, the rising of software-defined networking (SDN) and artificial intelligence (AI) makes it almost inevitable to combine these two promising technologies for an unprecedented level of network automation [1]. This is especially true, when the network to be managed is a multilayer one that carries dynamic and irregular traffic from a huge volume of network services, whose quality-of-service (QoS) demands are various, *e.g.*, a packet over elastic optical network (IPoEON) [2]. Therefore, people have designed a few machine learning (ML) based network control and management (NC&M) schemes [3, 4] to facilitate AI-assisted network automation (AIaNA) in software-defined IPoEONs (SD-IPoEONs). Through data analytics, these schemes can forecast the network status in an SD-IPoEON precisely and then make NC&M decisions accordingly. Hence, network resources can be automatically allocated/adjusted in advance to improve cost-effectiveness significantly. However, after celebrating the initial success of AIaNA, we should still be cautious about implementing it in production networks. This is because it is still unknown whether or not the reduction of human involvement achieved by AIaNA could bring unexpected reliability and security issues. More specifically, whether AIaNA can be fully trusted, to what extent it can be trusted, and whether it can completely replace a human operator, are unexplored questions. This inspires us to study how to mislead a human-free AIaNA system in this work.

We consider an SD-IPoEON that leverages ML-based traffic prediction to achieve AIaNA. Note that, in addition to traffic volume, AIaNA can also utilize ML to predict other parameters, such as quality-of-transmission (QoT), resource usage, and exception occurrence [3]. Nevertheless, it is known that such ML-based predictors are vulnerable to well-crafted adversarial samples [5]. Specifically, a malicious party can easily mislead an ML-based predictor to output incorrect predictions by mixing adversarial samples in its input time series. Although such adversarial-sample-based attacks can be addressed by introducing transfer learning in ML-based predictors [6], we argue that if the attacker is smart enough to generate the adversarial samples adaptively, the predictors could still be misled to make AIaNA unreliable. Therefore, we leverage deep reinforcement learning (DRL) to design an adversarial module (ADVM) that can craft and inject adversarial traffic samples adaptively to mislead the ML-based traffic predictor in an SD-IPoEON.

We show that the ADVM can monitor and interact with the SD-IPoEON to train its deep neural networks (DNNs) on-the-fly, such that adversarial traffic samples can be generated and injected in the most devastating and hard-to-detect manner. Then, AIaNA in the SD-IPoEON will be misled to make the multilayer service provisioning unreliable, *i.e.*, there will be unnecessary congestions/under-utilizations on lightpaths, and abnormal network reconfigurations will be

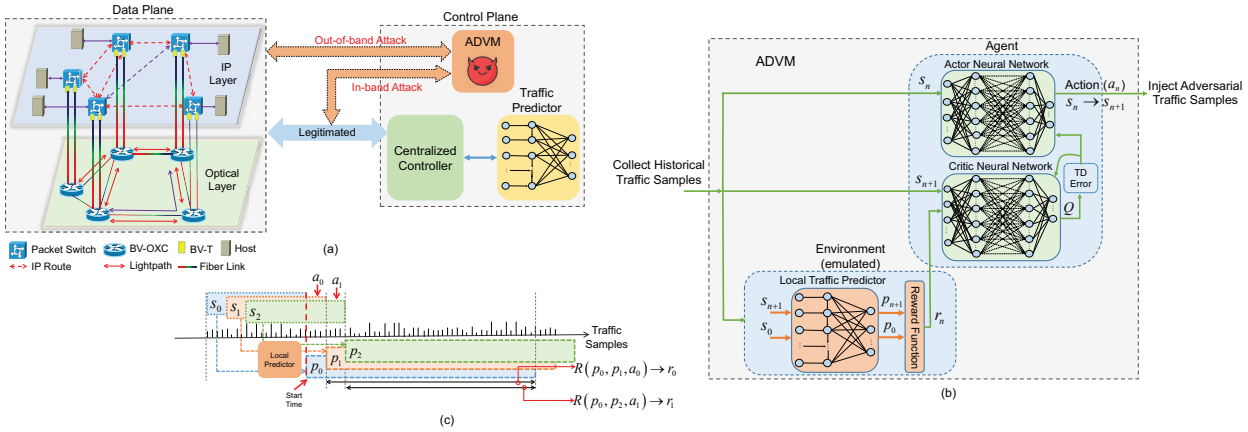

Fig. 1. System architecture, (a) SD-IPoEON with ADVM, (b) Design of DRL-based ADVM, and (c) Operation of ADVM.

invoked frequently to cause extra operational cost and complexity. Our simulations also study the tradeoff between the strength of perturbation due to adversarial samples and the impact of the adversarial-sample-based attack on AIaNA.

2. DRL-based Adversarial Traffic Sample Generation and Injection

Fig. 1(a) shows the architecture of an SD-IPoEON and ADVM's position in it. The data plane consists of IP and optical layers. The optical layer is built with a few bandwidth-variable optical cross-connects (BV-OXCs), which are interconnected by fiber links and can switch lightpaths with flexible-grid spectrum assignments [7]. On each lightpath, there is dynamic traffic from the IP layer, which is generated by the hosts, and is groomed and routed on the lightpath by the packet switches. Every packet switch connects to a local BV-OXC through a few bandwidth-variable transponders (BV-Ts), each of which can generate/terminate the optical signal of a lightpath. All the data plane elements are managed by the centralized controller. This means that the controller can install flow-tables in switches in the IP layer to groom and route IP flows on lightpaths, and it also can configure the BV-Ts and BV-OXCs in the optical layer to establish, reconfigure and remove lightpaths to adapt to dynamic IP traffic. Meanwhile, the controller monitors traffic condition, leverages ML-based traffic prediction to detect congestions/under-utilizations on lightpaths in advance, and adjusts multilayer service provisioning in the SD-IPoEON accordingly to achieve AIaNA. In this work, we assume that the ML-based traffic predictor is based on the long/short-term memory based neural network (LSTM-NN), since it is one of the most-used ML schemes for forecasting time series. Based on traffic prediction and current network status, the controller calculates future multilayer provisioning schemes with the CRV algorithm in [4].

On the other hand, as shown in Fig. 1(a), ADVM also resides in the control plane to monitor traffic condition for generating and injecting adversarial traffic samples adaptively. It can launch adversarial-sample-based attacks in either the in-band or out-of-band manner. For the in-band manner, ADVM taps and hacks the communications between the control and data planes, to collect legitimated traffic and inject adversarial samples. For the out-of-band way, it deploys a few traffic monitors in the SD-IPoEON for passive monitoring, and controls several hijacked hosts to inject adversarial traffic samples when necessary. We would not specify how the attacks are conducted here, since our ADVM can mislead the AIaNA in the SD-IPoEON in both manners. The detailed design of ADVM is shown in Fig. 1(b). It collects historical traffic samples regarding one or more lightpaths, and crafts the adversarial samples to inject accordingly. This is realized by letting the DRL agent based on the advantage actor critic (A2C) interact with the emulated environment provided by the local traffic predictor. The local traffic predictor in ADVM mimics the legitimated one that is attached to the controller. Note that, the two predictors do not have to use the same architecture.

Fig. 1(c) explains the operation of ADVM. The state observed by the DRL agent is a series of the most recent traffic samples collected from the SD-IPoEON. If we select an instant as the start time of ADVM's operation, the latest historical samples (for the traffic before the start time) constitute the initial state s_0. Using s_0 as the input, the local predictor forecasts the undisturbed future samples as p_0, which should describe the legitimated traffic accurately, *i.e.*, mimicking the legitimated predictor connected to the controller. Then, the agent determines its action a_0 on how to inject adversarial samples based on s_0, with its actor neural network (A-NN). More specifically, the action a_0 indicates when and how to disturb the legitimated traffic within a preset future duration. After the action having been applied, the observed state gets transferred to s_1, based on which the local predictor obtains a new prediction p_1.

Next, ADVM calculates the reward of the last action (r_0) with p_0, p_1, and a_0. Here, the reward increases with the average relative error between p_0 and p_1 for corresponding samples, and it decreases with the perturbation strength caused by a_0 on s_1. That is to say, the agent is trained to find the way that can use the smallest perturbation strength to mislead the legitimated predictor to give the largest prediction error. The agent gets trained in the online manner, which means that the aforementioned procedure will be repeated during network operation until the duration of p_0 has expired and a new p_0 is generated to continue. Through the process, the critic neural network (C-NN) in the agent evaluates the actions from the A-NN, and outputs the action-state value function (*i.e.*, the Q function) for calculating

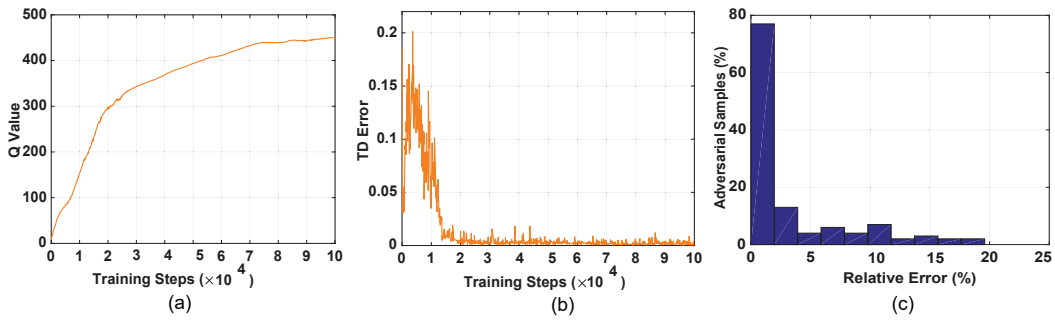

Fig. 2. Training performance of ADVM, (a) Q value, (b) TD error, and (c) Distribution of relative errors.

the temporal difference (TD) error. The TD error is then leveraged by the agent to update the parameters of its A-NN and C-NN. Meanwhile, it is easy to see that as long as the local traffic predictor uses transfer learning, ADVM will be able to craft adversarial samples in situations where the characteristics of traffic are time-varying.

3. Simulation Setup and Results

Our simulations use the 14-node NSFNET [4] as the optical layer topology. In the IP layer, dynamic flow requests are generated according to the Poisson process, while the legitimated samples of each flow follow realistic traffic traces [8]. In the simulations, ADVM can only modify at most 40% of the original samples, to launch its adversarial-sample-based attacks, while the actual samples to modify and the change made by each adversarial sample are determined by its DRL agent. Meanwhile, we assume that compared with its original one, each adversarial sample can only increase the traffic volume. This is because decreasing of traffic volume is not feasible in the out-of-band attacks. For a lightpath whose traffic is described by 160,000 time-varying samples, the performance of ADVM's online training is shown in Fig. 2. Figs. 2(a) and 2(b) suggest that the training converges quickly after ∼20,000 steps, since the TD error tends to be 0 and the Q value increases slowly thereafter. Then, if we limit the maximum relative error (RE) made by each adversarial sample as 20%, Fig. 2(c) plots the distribution of the REs. We observe that 90.9% of the adversarial samples have a RE below 5%, while the average RE is 3.2%. Hence, the changes made by the adversarial samples would be hard-to-detect.

Next, we attach ADVM to each established lightpath in the SD-IPoEON, let it launch adversarial-sample-based attacks with different perturbation strengths, and check its adverse effects. Here, we define the strength of perturbations made by adversarial samples as the maximum RE caused by each of them, and change its value from 5% to 20%. Fig. 3 summarizes ADVM's adverse effects on the AIaNA of the SD-IPoEON, where each data point is obtained by averaging the results from 10 independent runs, to ensure sufficient statistical accuracy. It can be seen that compared with the one without ADVM, the operation of the SD-IPoEON with it gets disturbed significantly. Specifically, the major metrics on congestions on lightpaths, bandwidth allocations, and network reconfigurations are all increased substantially, when the attacks present. Also, the increments become larger when the strength of perturbations increases. Hence, the results in Fig. 3 confirm that ADVM causes extra and unnecessary operational cost and complexity.

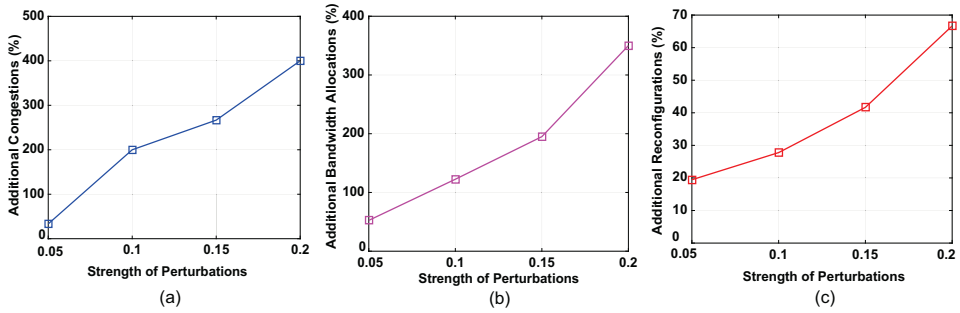

Fig. 3. Adverse effects of ADVM on (a) Congestions, (b) Bandwidth allocations, and (c) Network reconfigurations.

Finally, we check how ADVM performs when the characteristics of traffic are time-varying. More specifically, we make the legitimated traffic samples switch between two sets of traffic data, whose characteristics are different, and set the strength of perturbations as 10%. For such setting, ADVM still causes adverse effects, which are 100% additional congestions, 58% additional bandwidth allocations, and 12.5% additional reconfigurations. The results verify that ADVM is smart enough to generate and inject adversarial traffic samples adaptively. Compared with those in Fig. 3, the adverse effects are smaller. This because ADVM needs to adjust itself to adapt to the traffic condition changes.

4. Summary

By leveraging DRL, we proposed the ADVM that can craft and inject adversarial traffic samples adaptively to mislead the ML-based traffic predictor in an SD-IPoEON. Simulation results demonstrated its effectiveness.

References

[1] A. Mestres *et al.*, "Knowledge-defined networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, pp. 2-10, Jul. 2017.
[2] P. Lu *et al.*, "Highly-efficient data migration and backup for Big Data applications in elastic optical inter-datacenter networks," *IEEE Netw.*, vol. 29, pp. 36-42, Sept./Oct. 2015.
[3] D. Rafique *et al.*, "Machine learning for network automation: Overview, architecture, and applications," *J. Opt. Commun. Netw.*, vol. 10, pp. D126CD143, Oct. 2018.
[4] S. Liu *et al.*, "DL-assisted cross-layer orchestration in software-defined IP-over-EONs: From algorithm design to system prototype," *J. Lightw. Technol.*, vol. 37, pp. 4426-4438, Sept. 2019.
[5] N. Papernot *et al.*, "The limitations of deep learning in adversarial settings," *in Proc. of IEEE EuroS&P 2016*, pp. 372-387, Mar. 2016.
[6] J. Guo *et al.*, "When deep learning meets inter-datacenter optical network management: Advantages and vulnerabilities," *J. Lightw. Technol.*, vol. 36, pp. 4761-4773, Oct. 2018.
[7] Z. Zhu *et al.*, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing,"*J. Lightw. Technol.*, vol. 31, pp. 15-22, Jan. 2013.
[8] [Online]. Available: https://github.com/lsq93325/Traffic-creation/blob/master/README.md?tdsourcetag=s_pctim_aiomsg