JOURNAL OF
Optical Communications
and Networking

Coordination between control layer AI and on-board AI in optical transport networks [Invited]

Yongli Zhao,[1,*] Boyuan Yan,[1] Zhuotong Li,[1] Wei Wang,[1] Ying Wang,[2] and Jie Zhang[1]

[1]State Key Laboratory of Information Photonics and Optical Communications, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]State Grid Information & Telecommunication Company, Beijing 100761, China
*Corresponding author: yonglizhao@bupt.edu.cn

Received 8 July 2019; revised 26 August 2019; accepted 28 August 2019; published 16 October 2019 (Doc. ID 372055)

In optical transport networks, the urgent demand for control efficiency and intelligence has become one of the most significant challenges for telecom operators. With the development in control technology, more attention has been paid to performance enhancement of the centralized controller of software-defined optical networks (SDONs). Meanwhile, machine learning (ML) is emerging as a promising technology to facilitate the intelligence of control planes in SDONs. Some research works have been conducted to use ML to solve problems in optical transport networks. However, it is still a challenge to deploy and use computing resources. On the one hand, computing resources can be deployed inside the centralized controller of an SDON to enable control layer artificial intelligence (AI). On the other hand, computing resources can also be deployed on the hardware board to enable on-board AI. The two-layer AI functions are able to meet different intelligent requirements in data and control layers in different scenarios. Therefore, coordination between them is an important issue. In this paper, a novel control architecture based on an SDON is proposed, and it can support control layer AI and on-board AI simultaneously. Particularly, on-board AI is proposed based on edge computing to support various ML applications. To evaluate the proposed architecture, we develop an experimental testbed and demonstrate a typical use case, i.e., alarm information prediction. Experimental results show that coordination and cross-layer optimization between control layer AI and on-board AI can be achieved. However, there is much space for research in this area, and we envision some open issues.

https://doi.org/10.1364/JOCN.12.000A49

1. INTRODUCTION

With the breakthrough of artificial intelligence (AI) in recent years [1,2], AI has been applied in many fields, such as computer science, finance, trade, medicine, diagnosis, heavy industry, transportation, etc. As a representative technique of AI, machine learning (ML) has become a hot topic recently. ML was initially studied in 1988 [3], mainly targeting pattern recognition and computing learning theory. Now, ML is being widely used in various fields. In the field of image classification, for example, ML achieves and further exceeds human-level recognition capability [4–6]. In gaming, AlphaGo [2], developed by Google's DeepMind Research Group, beat the world's Go champion. Meanwhile, ML has also recently been used in data mining [7] and natural language processing [8].

In optical transport networks, ML is emerging as an advanced tool for dealing with complex issues in two different perspectives. For optical transmission, ML is capable of estimating fiber linear/nonlinear impairment according to signal and device parameters [9]. For example, both the unsupervised and supervised ML methods are applied to improve performance of the optical communication system based on the nonlinear Fourier transform [10]. ML can also be used to estimate the nonlinear noise and thereby monitor the optical signal-to-noise ratio (OSNR) [11]. Fiber-induced intra- and inter-channel nonlinearities are tackled using blind nonlinear equalization by unsupervised ML-based clustering in ∼46 Gb/s single-channel and ∼20 Gb/s multichannel coherent multicarrier signals [12]. The pre-distortion scheme based on a clustering algorithm of ML can mitigate nonlinear impairments for a VLC system [13]. For optical networks, ML can deal with issues such as traffic load prediction, failure location, and resource allocation [14]. For example, the network management system can flexibly manage resources in the backbone IP/optical network by using machine learning to predict traffic flows in the short- and long-term [15]. A deep-learning-based failure prediction algorithm is proposed, which constructs the data set based on data augmentation for data training [16]. A resource-allocation method based on

reinforcement learning is proposed for multimodal optical networks [17]. Besides, ML can also detect intrusions in the control plane of a software-defined optical network (SDON) [18]. Simulation results show that the accuracy of an intrusion detection scheme can reach more than 85%. The research shows that ML can achieve good performance in dealing with different problems in optical networks.

However, ways to deploy AI functions in optical transport networks remain an open issue. Specifically, how to implement training, testing, and application of the AI model are three major issues. First, in the training process, the ML engine needs to be fed with a large amount of data, which means that the components that have ML capability require storage resources to save data sets and computing resources to update the parameters of the models (such as the forwarding transmission of the neural network). Second, for the testing process, because the testing data set is much smaller than the training data set, the testing process does not require too many storage and computing resources. Finally, in some cases, the AI component should have the ability to make real-time responses. Therefore, AI modules are usually located on a resource-rich device such as the central controller of an SDON, which can be considered as control layer AI. However, the AI module of the central controller part cannot deal with the local problems of each network element efficiently unless all equipment can report their local data to the central controller in real-time. Because such many-to-one synchronization will incur a heavy workload to the central controller, the controller side AI is not an ideal place for solving equipment-side problems. Thus, it is necessary to deploy AI capability into the data plane in a distributed way. Up to now, few works have studied how to deploy AI functions in optical transport nodes to enable equipment-level applications.

This paper introduces the concept of deploying on-board AI into optical networks to enable AI in transport network equipment. The architecture of on-board AI is designed based on edge computing, and the collaboration mode between control layer AI and on-board AI is studied. Control layer AI is deployed with an SDON controller, and on-board AI is deployed with optical transport equipment. An experimental testbed is built to evaluate performance between the two-layer AI. On-board AI is performed on a circuit board with an AI-specific chip, along with other necessary components like memory, express card, etc. We also study the potential applications based on on-board AI, even the coordination between control layer AI and on-board AI. This paper is an extension of [19]. The major new contribution of this paper is that we describe the process of training/testing the AI model and the method of selecting the best model from the AI model library in the collaboration mode between control layer AI and on-board AI in more detail. Meanwhile, we develop an experimental test platform for evaluating the proposed collaboration mode. The experimental test platform takes the ML-based traffic prediction function as an example. The embedded AI-board DP-8020 is used as the on-board AI to evaluate the efficiency of the training/testing model process in the collaboration mode. The experimental results prove that the proposed collaboration mode can improve the efficiency of AI control functions in optical networks.

The rest of this paper is organized as follows. In Section 2, we introduce the architecture of self-optimized optical networks (SOONs), which is designed based on SDONs and supports AI. Detailed architecture is also given to show how the two-layer AI is deployed. In Section 3, we introduce the collaboration mode between the control layer AI and on-board AI. In Section 4, we perform experiments in two scenarios and discuss the results. We first build an experimental testbed to evaluate the network architecture we proposed. We also demonstrate a typical use case. Section 5 discusses the open issues in the proposed network architecture, and Section 6 offers a conclusion.

2. NETWORK ARCHITECTURES

A. Software-Defined Optical Networks

SDONs introduce software-defined networking (SDN) into optical networks to achieve unified scheduling and control capability over various resources in the optical layer [20]. As shown in Fig. 1, this control architecture can abstract all the lower-layer resource information as general application programming interface (API) functions for the upper-layer applications, such as port discovery, resource collection, path computation, connection creation and deletion, etc. In the architecture, the data layer consists of the transport equipment controlled by SDON controllers that are located in the control layer. Each local controller manages a single domain. The orchestration layer is located between the application layer and control layer, which are close to the operators and allow network operators to define their network through software configuration files or policies that are written in a language that the control layer can understand. The main component of the orchestration layer is the network orchestrator, which is used to connect controllers in different domains and collect the status information of the data layer. Instead of setting network services and deploying applications separately, the orchestrator can automate workflows, so that the operations on services and applications can be instantiated automatically. The transport API (T-API) in an SDON controller realizes the control virtual network interface (CVNI) between the orchestration and control layers. The transport also ensures that the data processing in the function bundles is independent

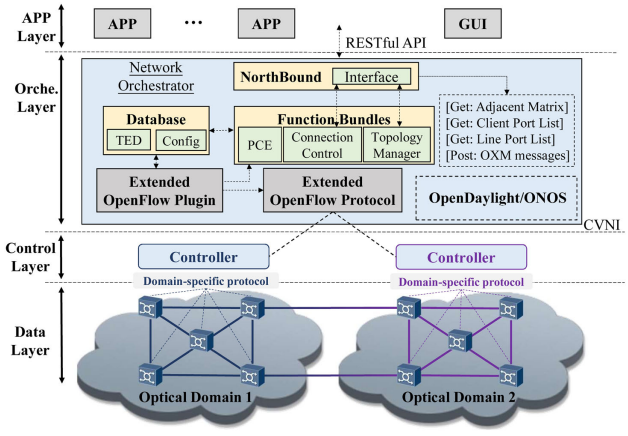

Fig. 1. SDON architecture.

of the communication protocol. The database module collects topology, ports, and configuration information from each optical domain. Here, each domain abstracts local topology as a smaller aggregated topology for the orchestrator. Based on the open-source Open Services Gateway initiative framework, a topology manager, path computation element, and connection control are developed as the function bundles. As a typical northbound interface protocol, a representational state transfer (RESTful) API can be used over nearly any protocol and allows developers to build an API that meets the needs of diverse applications for different customers.

B. Self-Optimizing Optical Networks

Based on the above SDON architecture, we proposed a novel optical network architecture, i.e., SOON, which can integrate ML technology into an SDON to improve the intelligence of optical networks [21,22]. As shown in Fig. 2, the architecture of an SOON consists of a data layer, model layer, and policy layer. The data layer contains physical/virtual optical networks. Each element of the networks is named the self-optimizing network element (SNE), which connects to a model layer via a management and data interface (MDI). The MDI consists of two types of protocols and a unified network model on them. Network control protocols (NCPs) are traditional network protocols. The SNE reports network status and receives instruction through the NCP. The main purpose of status-aware protocols (SAPs) is to enable the model layer to perceive detailed information of network elements about physical components, for example, physical parameters like the OSNR and environment parameters like temperature. Such an elaborative data-acquisition method defined by SAP aims to provide mass data in as many dimensions as possible for ML applications. Besides, SAP should also provide direct control capability for the model layer to adjust parameters in the SNE. Currently, there are no SAP standards, although there is a related draft to define a SAP [23]. In the MDI, the unified network module aims to provide a protocol-free interface for the network control core (NCC) and machine learning engine (MLE), which could gather and filter all information from NCPs and SAPs and reformat these data by following some unified network model format.

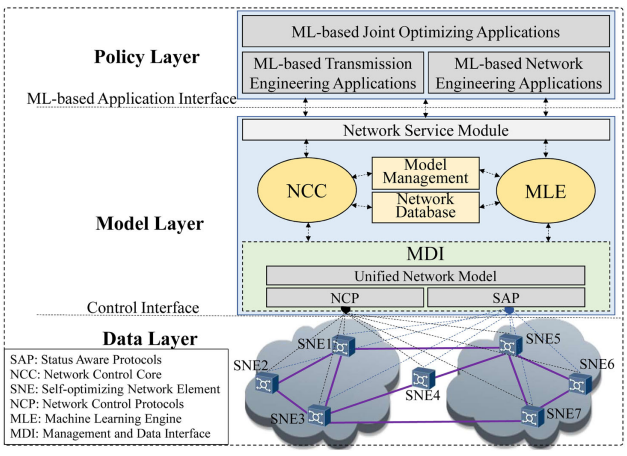

Fig. 2. Self-optimizing optical networks [21].

The NCC provides the traditional basic functions of an SDN controller. The MLE is a built-in ML-enabled framework, which includes an ML algorithm library and a related lifecycle control interface. The algorithm library provides various ML algorithms. In the model layer, two modules connect the NCC and MLE indirectly. The network database module is the source of all the data in the model layer, including the traffic engineering database (TED), which stores the network traffic performance data from the data layer. The NCC can synchronize the network state from the network database and the MLE can obtain the training/test/verification data set. The model management module is a well-trained model repository responsible for creating, deleting, replacing, and searching for ML models. The NCC invokes the model according to the requirements of the policy layer and obtains the final model from the model management module. The MLE executes the instructions from the upper layer through the network service module and operates on the specific data set accordingly. The network service module provides a unified interface for the policy layer, called the ML-based application interface, which enables open and secure access to the NCC and MLE.

All ML-based applications are placed in the policy layer. These applications execute specific policies with the help of the ML model and control network operations through the NCC. Besides the applications about network engineering, there also exists ML-based transmission engineering applications and ML-based joint optimizing application.

In the context of an SDON, the controller has sufficient resources and data to train the AI engine. In the control plane, well-trained AI models can help network operators to do the work that relies heavily on human experience. The AI engine is the key technology of the intelligent control layer, but deploying AI only in the controller is not enough to build a fully automated network, as the data layer also has heavy work that relies heavily on experience. On the one hand, the AI engine in the control layer can handle networking-level issues naturally, but it is difficult to handle equipment-level issues in the data layer due to the lack of data from transport network devices. On the other hand, it will be a heavy burden on the centralized control layer AI engine if all the data of the data plane flows to the control layer for multi-instance AI-related processing. To fully utilize the ability of AI for building network automation, this paper proposes the concept of on-board AI for optical networks, which refers to embedded AI in the optical transport equipment of the data layer. Figure 3 illustrates the architecture of the transport equipment and controller with AI for optical networking. A detailed description of the implementation of control layer AI and on-board AI is as follows. A detailed description of their collaboration mode is given in Section 3.

C. Control Layer AI

It is common sense that AI can be part of the control layer [as shown in Fig. 3(a)], and it can make networking-level optimizations based on the controller-scope data, which is usually from a TED. Due to the huge demand of computing and storage resources, control layer AI can be embedded in an SDON controller with rich graphics processing unit (GPU)

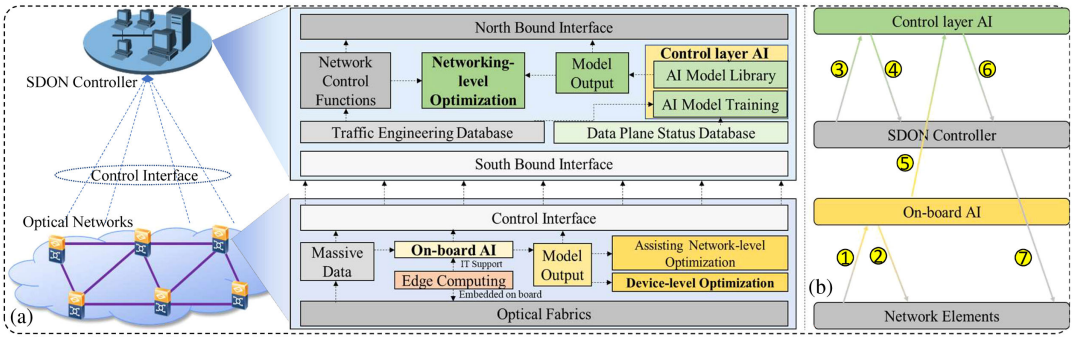

Fig. 3. Architecture of on-board AI for optical networking. (a) Network architecture. (b) Workflow of the AI engines.

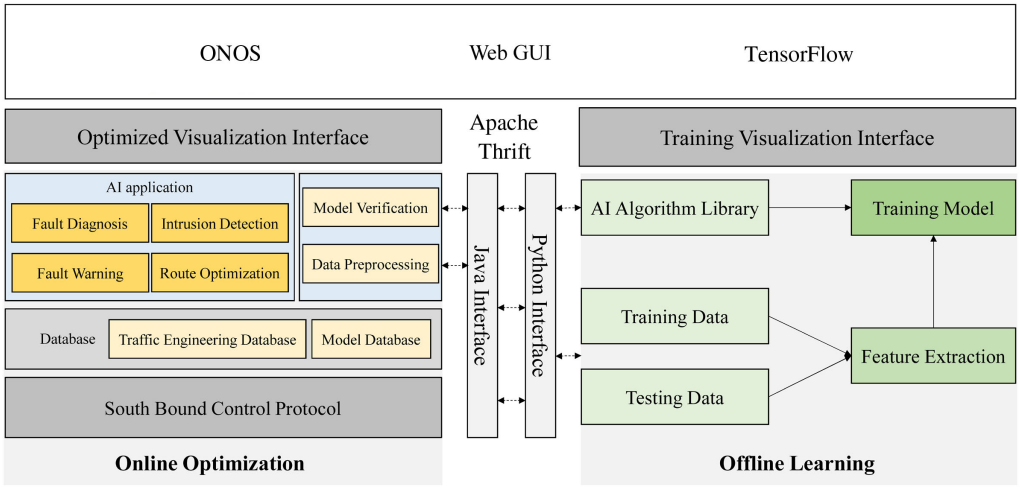

Fig. 4. Offline training and online optimization in SOON platform.

resources or be connected to an SDON controller based on an open-source software platform, such as TensorFlow.

The architecture of the SOON platform based on an ONOS controller and TensorFlow is shown in Fig. 4. The platform selects the AI algorithm from the AI algorithm library according to the requirements of the AI application and trains and tests the AI model using the training/test data offline. Then, the ONOS controller selects and verifies the trained model from the model database and achieves the corresponding AI application functions online. In this way, offline training learning and online optimization can be combined to save the consumed resources of the SDON controller.

However, with on-board AI, the control layer requires certain transformation to support cross-layer AI coordination. In detail, the controller should be able to talk with the data layer on AI-related data and indicators. Thus, we introduce a data plane status database (DPSD) into the control layer to store the data from on-board AI, as shown in Fig. 3(a). With DPSD, the centralized AI engine can make cross-layer networking-level optimizations by considering equipment-level status.

D. On-Board AI

With a centralized control layer in optical networks, the transport equipment in the data layer is simplified just for data forwarding and thus lacking the ability for running AI engines. Hence, for deploying AI in the data layer, we first introduce edge computing into transport equipment to enhance its capability for AI processing and data storing. Edge computing resources can be deployed together with the transport equipment, as shown in Fig. 3(a). With edge computing, AI engines can be embedded in each piece of transport equipment in an on-board fashion. Located inside the equipment, the on-board AI engine has access to all the data of the local equipment, including the performance statistics (e.g., the laser power of transponders, the noise level of each port, etc.) and running status of each component (e.g., temperature and running time of optical fabrics, historical failure records, etc.). With such equipment-scope data, the on-board AI can benefit optical networks in two ways, i.e., 1) solving equipment-level issues and making local optimization and 2) extracting indicators from the original data for assisting network-level optimization by sending indicators to controllers.

Compared with the traditional transport equipment, AI-embedded equipment owns additional components to enable AI processing functions. Figure 5 shows the AI board diagram to enable on-board AI. All key features of AI applications are extracted from the data set. Traditional equipment performance like the output optical power is not enough to analyze the equipment in detail. Then, equipment-level optimization needs more types of performance in equipment, and more

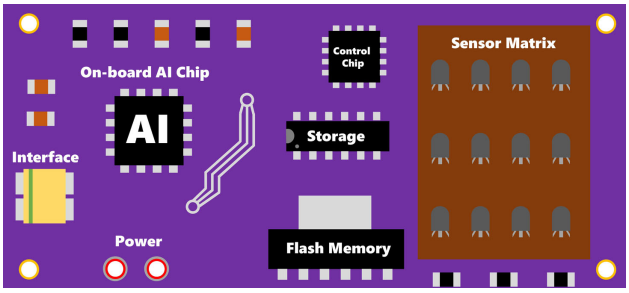

Fig. 5. Board diagram of on-board AI.

sensors for real-time performance monitoring and data collection are embedded in the equipment at first. There is a sensor matrix composed of many sensors on the AI board in Fig. 5, and the AI board carries a low-power AI chip as the intelligent core and storage to save continuous performance data. Besides, control chips and interfaces are also deployed on the board to complete task management and connect with other boards and controllers, respectively. Finally, one piece of transport equipment should be able to carry multiple AI boards to enable scalable and flexible data processing ability.

3. COLLABORATION MODE BETWEEN CONTROL LAYER AI AND ON-BOARD AI

Table 1 compares the features and performance of the control layer AI and on-board AI. The performance of on-board AI is summarized from multiple vendors of AI-enabled boards (for example, Xilinx, Cambricon Technologies, Horizon Robotics, etc.). The embedded AI board has the characteristics of low power consumption and low price, which also limits its computing power. The AI-enabled board uses a field programmable gate array (FPGA) to effectively run the artificial intelligence model usually, which is a common method to improve the computing power for specific AI-related tasks. Currently, although both of them cost similar time in the embedded field, most of the AI-enabled boards carry inference-only chips, which have limited computing resources and can only accelerate the model inference (for example, forwarding propagation in neural networks) but cannot handle parameter updating. In other words, most on-board AI only supports model testing instead of model training. In this paper, the AI-enabled board is designed to be inserted into the extensible slot of optical transport equipment (for example, an optical transport network). Hence, for equipment-level data processing, the on-board AI can provide a faster response than the remote cloud controller. In addition, in this paper, on-board AI is deployed in a distributed form on several different nodes in the data layer of optical transport networks, so it has stronger survivability than an SDON controller natively.

AI can be applied for solving equipment-level and networking-level issues, respectively. Figure 3(b) shows the data flow of an AI application in different cases. In the data plane, the on-board AI can 1) collect data from equipment and make decisions for optimizing or managing local equipment. In the control plane, the control layer AI can collect data from the controller and make decisions for optimizing or managing network services. In addition, the control layer AI and on-board AI can coordinate to solve cross-layer issues by taking the output of on-board AI and the data from the controller into consideration and make decisions for managing network services and further managing equipment.

Table 1. Comparison between Control Layer AI and On-Board AI

Evaluation	Control Layer AI	On-Board AI
Model training	Supported	Not supported
Model testing	Supported	Supported
Computing resource	Huge	Small
Power consumption	High	Low
Delay for network-level application	Low	High
Survivability	Weak	Strong
Price	High	Low

In the AI algorithm, the hyper-parameter is one of the key points, which has an important influence on the performance of the model, but there is still no effective solution. In addition, in order to achieve a better optimization effect, it is usually necessary to try the combination of multiple hyper-parameters and select the best model from the trained models. This requires a lot of computing resources, even beyond the capabilities of control layer AI. To solve this problem, this paper proposes a universal AI collaborative model selection mode, as shown in Fig. 6. In the central controller, the control layer AI is composed of numerous AI engine groups that can jointly perform network-level optimization. The model database can store AI models and provide rapid external access. After each epoch or specific iteration, the model needs to be saved on the database. Then, the on-board AI downloads the model and executes testing on the testing data set. In order to run the model on an FPGA, there should be at least three steps to translate it normally, i.e., compression, compilation, and runtime [24]. In the compression step, the model is pruned to filter useless parameters and then quantized to convert 32/64-bit float/double parameters to unsigned/signed 8 bit integers with very small accuracy degradation. The compilation step is used to convert the model into efficient FPGA instructions through a compiler, assembler, and so on. FPGA only handles the core computing of the AI model, and the runtime step is used to download the input and calculate the loss value from the data set and model's output.

Training on control layer AI and testing on on-board AI could run in semiparallel. When multi-epoch training with the same hyper-parameters is finished, the board would select the best model locally according to the testing results. If multiple training processes with different hyper-parameters run simultaneously, there should be multiple boards to test the model jointly. In addition, the control layer AI needs to select the global model after the local model selection on different boards. Finally, according to the application type, the selected best model is deployed on the SDN controller or optical transport equipment directly.

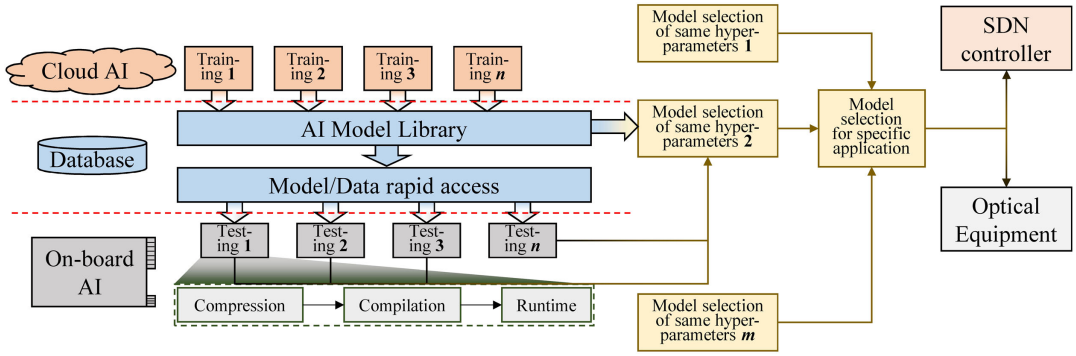

Fig. 6. Universal collaboration mode for joint optimization in optical transport networks.

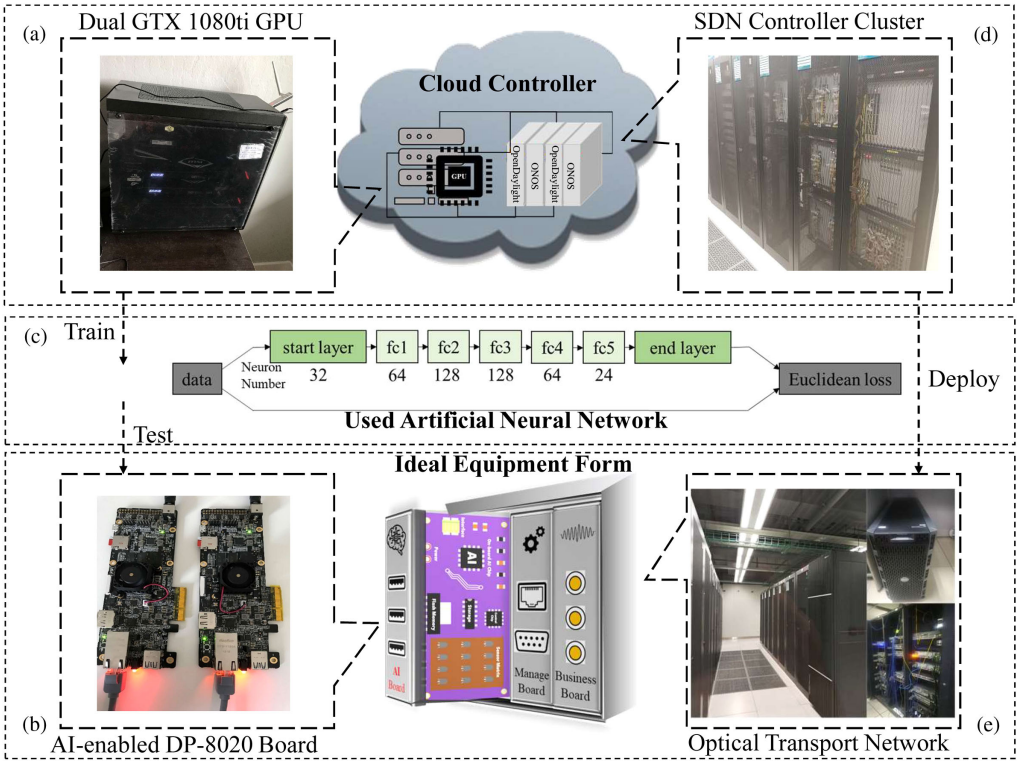

Fig. 7. Testbed: (a) control layer controller, (b) AI-enabled DP-8020 board, (c) used artificial neural network, (d) SDON controller cluster, (e) deployed optical networks.

4. EXPERIMENTAL SCENARIO AND RESULTS

A. Evaluation of the Collaboration Mode between Control Layer AI and On-Board AI

In order to evaluate the efficiency of the proposed collaboration mode in the SOON architecture, an experimental testbed for traffic prediction in metro transport networks is developed, as shown in Fig. 7. The data source comes from a private ISP with centers in 11 European cities. The data corresponds to a transatlantic link and was collected every 5 min from 06:57 on 7 June to 11:17 on 31 July 2005 [25]. After preprocessing, the sizes of the training data set and testing data set are 15,904 and 4000, respectively. Figure 7(a) shows a high-performance computer with two powerful GTX1080Ti GPUs, on which the control layer AI is deployed in our testbed. Figure 7(b) shows two embedded circuit boards named DP-8020 that enable on-board AI locally. The GPU machine and the embedded boards are connected with each other through the Ethernet cables. All models would run on DP-8020's deep learning processing unit (DPU) after compression, compilation, and runtime steps. Figure 7(d) shows a cluster of ONOS-based SDON controllers. The controllers and the GPU machine negotiate with each other to control the network intelligently. Figure 7(c) shows a six-layer artificial fully connected neural network (FCNN) where Euclidean loss calculation is used to predict future traffic trends. After model selection, the best model will be deployed in the optical networks, as Fig. 7(e) shows. The fully connected neural network consists of an input layer, a hidden layer, and an output layer. Each layer consists of multiple neurons. The input of the FCNN is the sample data,

Table 2. Performance Sheet

	DP-8020	GTX-1080Ti
Power	15 W	80 W
Training time of each epoch	/	784.98 ms
Testing time of a model	1016.60 ms	98.12 ms

and the output is the predicted target. The fully connected neural network means that any neuron of the $n-1$ layers is connected to all neurons of the n layer. The neuron number of middle layers of the start layer, fc2-5 layer, and end layers are 32, 64, 128, 128, 64, and 24, respectively. The batch size is used to control the one-time throughput while training, which is set as 16. Therefore, each epoch needs $15{,}904/16 = 994$ iterations. The total number of epochs is set as 100, which is a commonly used setting. In order to evaluate the trained models as finely as possible, the model will be saved and tested on the AI board after every training epoch. The output neuron number is 24, which means the model can predict 24-time points in the future, i.e., 2 h. The input is composed of two parts. The first part contains three neurons that specify the day of the week, the hour of the day, and the minute of the hour, respectively. In addition, the second part specifies traffic values in the past several hours for every 5 min. The number of input neurons and basic learning rates are two important hyper-parameters for model training. In this paper, we use them in combination to find the best model.

Table 2 shows the practical performance collected from the experiment. The power of DP-8020 is much less than GTX 1080Ti. Figure 8 shows the lowest loss value under different combinations with learning rate and input neuron number. Each value of the figure means the best model selected from 100 models, which are collected in each epoch while training. By comparing multiple best models in Fig. 8, the best global model is selected, where the learning rate equals 1e-3, and the input neuron number equals 27.

In order to evaluate the time efficiency by using on-board AI. We assume that the forwarding time per data item on the GPU is F, the backpropagation time per item is B, the forwarding time per data on board is f, and the size of training data and testing data are R and E. Thus, the time cost for the training

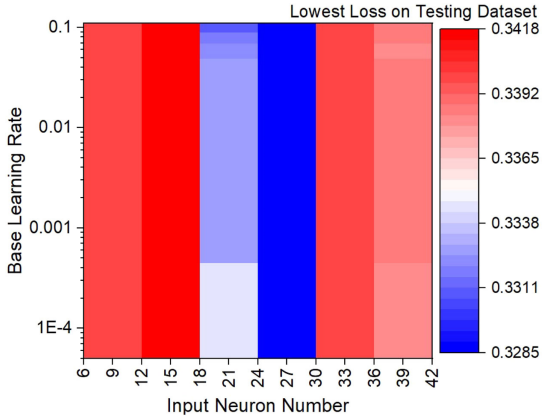

Fig. 8. Lowest loss for different learning rates and input neuron numbers.

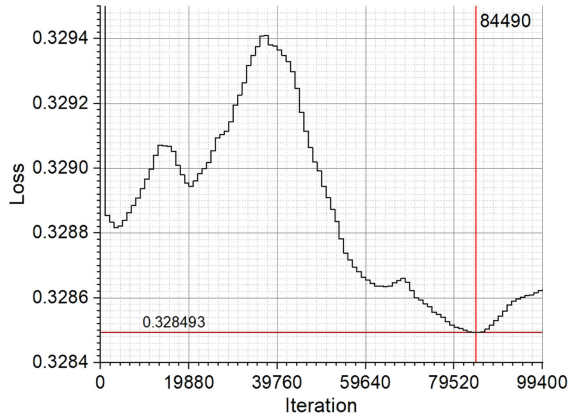

Fig. 9. Training process of the best model with the global lowest loss value.

process on the GPU, testing process on the GPU, and testing process on board are R/F, B/E, and f/E. If f/E < R/F, that means that the testing process could be deployed on board. Then, the efficiency improvement is (B/E)/(R/F).

The average time cost for an epoch on the controller (i.e., R/F) is about 784.98 ms, and the average time cost for a test on the AI board is about 1016.60 ms, as Table 2 shows. Because two boards are used concurrently, which halves the testing time cost as 508.30 ms (i.e., f/E), this time cost is lower than the training cost, so the testing phase is not blocked. If the test is located on the SDN controller, the average time for a test is 98.12 ms (i.e., B/E). Therefore, the efficiency is improved by $(B/E)/(R/F) = 98.12/784.98$, i.e., 12.5%. Besides, the training model is an unstable process. Figure 9 shows the loss value when training the best global model. After fluctuation, loss reaches the lowest value 0.328493 at the 84,490th iteration.

B. Demonstration of Alarm Prediction

Failures in optical networks will cause network service interruption, which may result in severe economic loss [19]. Alarms are indicators for identifying failures in optical transport networks. However, the number of alarms from all network equipment is too huge for an individual controller to deal with. Based on the fact that performance records are generated continuously, and most of the alarms are duplicated, the on-board AI engine in each piece of equipment can preprocess the original alarms and extract the indicators to improve the efficiency of failure processing.

Figures 10 and 11 show the benefits of alarm preprocessing based on on-board AI. They have access to all kinds of alarms and running parameters, including running temperature, transmitting power, receiving power, CPU utilization, etc. With the on-board AI engine, these data can be preprocessed locally. In each piece of equipment, one failure usually causes several chained alarms, and the root alarm is the key to identify the corresponding failure. In this case, on-board AI can help to filter the potential root alarms out from all alarms, and Fig. 10 compares the number of original alarms and filtered alarms. In addition, alarms are usually correlated with some

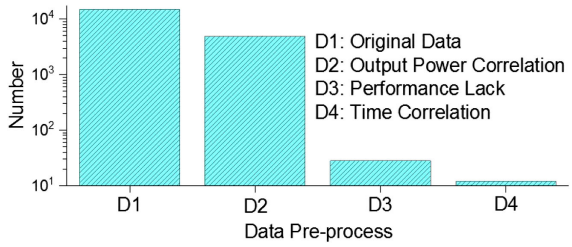

Fig. 10. Demonstration results: preprocessing of alarms.

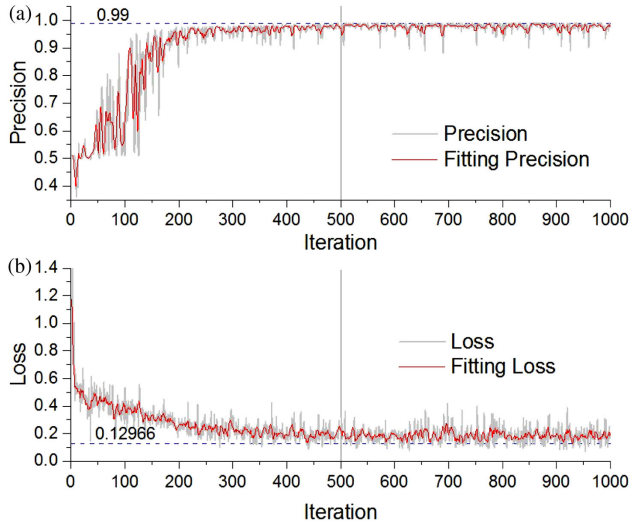

Fig. 11. (a) Prediction precision. (b) Training loss.

abnormal running parameters, which can also help to identify the failures. Thus, on-board AI can help to filter out the corresponding running parameters for each specific failure.

The accuracy rate is the ratio of the number of correctly classified data over the total number of input data. In general, alarm prediction is a binary problem. The output results indicate whether or not an alarm will happen, including yes and no labels. The precision of prediction in Fig. 11(a) is consistent with the actual value, which can reach 99% after multiple iterations. Loss value in Fig. 11(b) represents the gap between the predicted value of the model and the real value, and the loss value shows obvious convergence.

5. OPEN ISSUES

Coordination between control layer AI and on-board AI is necessary for future optical transport networks. However, there are some open issues for coordination.

A. Computing Resource Deployment

With the development of on-board AI and AI computing acceleration technology, on-board AI can support more open-source machine learning frameworks and take on more computing tasks. For example, the Jetson Nano recently launched by NVIDIA can offer 472 GFLOP and a variety of machine learning frameworks. The first open issue is how to deploy the computing resources between the control layer AI and on-board AI. Properly coordinating the distributed computing resources of the intelligent terminal and the centralized resources of the control layer AI (e.g., the controller cluster) can ensure the self-optimizing performance of the overall optical network. In the future, we will try to compare the impact of the performance of different types of AI board on the proposed network architecture.

B. Massive Data Collection

ML algorithms require massive data to ensure better learning and a predictive performance. This requires the optical network equipment to have a good ability to collect large amounts of data. Meanwhile, different device performance data are required for different machine learning algorithms. As more machine learning algorithms for optical networks are studied, it may be considered whether to collect network topology information. Hence, how to collect and process massive data is one of the next research areas of self-optimizing optical networks.

C. Application Scenario Selection

At present, our related work has some applications for ML algorithms such as failure location and traffic prediction. However, these ML algorithms are specific to certain application scenarios. Hence, the choice of ML algorithms for different application scenarios is an open issue. It is important to choose a specific ML model for a particular application scenario. This may require the AI model library to have a certain divisional difference between the corresponding models of different scenarios.

6. CONCLUSIONS

In this paper, we discuss on-board AI based on edge computing together with AI based on an SDON controller. Both should be conducted collaboratively to complete cross-layer optimization. We integrate on-board AI into optical transport networks to enable the deployment of equipment-level applications. Then, we propose a collaboration mode of joint optimization between control layer AI and on-board AI. The mode completes the data collection, ML algorithm model selection, and optimization through the interaction between the SDON controller and the device. The experimental results show that the computational efficiency of the model training is improved by 12.5% through the collaborative mode of the control layer AI and the on-board AI. Taking the ML application of the alarm prediction as an example, the correct prediction rate of the alarm in the network can reach as high as 99%.

Funding. National Natural Science Foundation of China (NSFC) (61571058, 61822105); Fundamental Research Funds for the Central Universities (2019XD-A05); State Key Laboratory of Information Photonics and Optical Communications of China.

REFERENCES

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *25th International Conference on Neural Information Processing Systems (NIPS'12)*, Lake Tahoe, Nevada, December 2012.
2. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," Nature **529**, 484–489 (2016).
3. D. N. L. Levy, *Computer Games I* (Springer, 1988).
4. R. Wang, B. Chen, J. Guo, and J. Zhao, "The image recognition based on restricted Boltzmann machine and deep learning framework," in *4th International Conference on Control and Robotics Engineering (ICCRE)*, Nanjing, China, 2019, pp. 161–164.
5. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, Massachusetts, June 2015.
6. C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *31st AAAI Conference on Artificial Intelligence (AAAI-17)*, San Francisco, California, February 2017.
7. J. Bellary, B. Peyakunta, and S. Konetigari, "Hybrid machine learning approach in data mining," in *2nd International Conference on Machine Learning and Computing*, Bangalore, India 2010, pp. 305–308.
8. H. Isahara, "Resource-based natural language processing," in *International Conference on Natural Language Processing and Knowledge Engineering*, Beijing, China 2007, pp. 11–12.
9. F. N. Khan, C. Lu, and A. P. T. Lau, "Machine learning methods for optical communication systems," in *Advanced Photonics*, New Orleans, USA, July 2017.
10. O. Kotlyar, M. Pankratova, M. Kamalian, A. Vasylchenkova, J. E. Prilepsky, and S. K. Turitsyn, "Unsupervised and supervised machine learning for performance improvement of NFT optical transmission," in *IEEE British and Irish Conference on Optics and Photonics (BICOP)*, London, UK, 2018, pp. 1–4.
11. A. S. Kashi, Q. Zhuge, J. Cartledge, A. Borowiec, D. Charlton, C. Laperle, and M. O'Sullivan, "Artificial neural networks for fiber nonlinear noise estimation," in *Asia Communications and Photonics Conference*, OSA Technical Digest (online) (Optical Society of America, 2017), paper Su1B.6.
12. E. Giacoumidis, A. Matin, J. Wei, N. J. Doran, L. P. Barry, and X. Wang, "Blind nonlinearity equalization by machine-learning-based clustering for single- and multichannel coherent optical OFDM," J. Lightwave Technol. **36**, 721–727 (2018).
13. X. Lu, M. Zhao, L. Qiao, and N. Chi, "Non-linear compensation of multi-CAP VLC system employing pre-distortion base on clustering of machine learning," in *Optical Fiber Communication Conference*, OSA Technical Digest (online) (Optical Society of America, 2018), paper M2K.1.
14. Z. Wang, M. Zhang, D. Wang, C. Song, M. Liu, J. Li, L. Lou, and Z. Liu, "Failure prediction using machine learning and time series in optical network," Opt. Express **25**, 18553–18565 (2017).
15. G. Choudhury, D. Lynch, G. Thakur, and S. Tse, "Two use cases of machine learning for SDN-enabled IP/optical networks: traffic matrix prediction and optical path performance prediction [Invited]," J. Opt. Commun. Netw. **10**, D52–D62 (2018).
16. L. Cuia, Y. Zhao, B. Yan, D. Liu, and J. Zhang, "Deep-learning-based failure prediction with data augmentation in optical transport networks," in *17th International Conference on Optical Communications and Networks (ICOCN2018)*, Zhuhai, China, November 2018.
17. B. Yan, Y. Zhao, Y. Li, X. Yu, J. Zhang, Y. Wang, L. Yan, and S. Rahman, "Actor-critic-based resource allocation for multi-modal optical networks," in *IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, United Arab Emirates, 2018, pp. 1–6.
18. H. Zhang, Y. Wang, Y. Zhao, and J. Zhang, "Exploring machine-learning-based control plane intrusion detection techniques in software defined optical networks," Opt. Fiber Technol. **39**, 37–42 (2017).
19. Y. Zhao, B. Yan, W. Wang, Y. Lin, and J. Zhang, "On-board artificial intelligence based on edge computing in optical transport networks," in *Optical Fiber Communication Conference (OFC)*, OSA Technical Digest (Optical Society of America, 2019), paper Tu2E.1.
20. Y. Ji, J. Zhang, and Y. Zhao, "Development prospects of software defined optical networks," Telecommun. Sci. **30**, 19–22 (2014).
21. Y. Zhao, B. Yan, D. Liu, Y. He, D. Wang, and J. Zhang, "Self-optimizing optical networks with machine learning," Opt. Express **26**, 28713–28726 (2018).
22. B. Yan, Y. Zhao, W. Wang, L. Yan, Y. Wang, J. Liu, S. Zhang, D. Liu, Y. Lin, H. Zheng, and J. Zhang, "First demonstration of machine-learning-based self-optimizing optical networks (SOON) running on commercial equipment," in *European Conference on Optical Communication (ECOC)*, Rome Italy, 2018, pp. 1–3.
23. B. Yan, Y. Zhao, W. Wang, X. Yu, and J. Zhang, "An equipment parameter control architecture," https://datatracker.ietf.org/doc/draft-epc-architecture/.
24. Xilinx Inc., "DNNDK: full-stack solution for deep learning development & deployment," http://www.deephi.com/technology/dnndk.
25. "Internet traffic data from an ISP," https://github.com/SeSaMe-NUS/Smiler/tree/master/data/isp.

Yongli Zhao (M'09, SM'15) was born in Hebei Province, China, in 1981. He received B.E. and Ph.D. degrees from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2005 and 2010, respectively. His major was electronic science and technology. He became an assistant professor in the Institute of Information Photonics and Optical Communications at BUPT in 2010. In Dec. 2014, he became an associate professor in the Institute of Information Photonics and Optical Communications at BUPT. During Jan. 2016 to Jan. 2017, he was a visiting associate professor at UC Davis. Since Dec. 2018, he has been a full professor in the Institute of Information Photonics and Optical Communications at BUPT. Up to now, he has published more than 300 international journal and conference papers. His current research focuses on machine learning in optical communication networks, quantum key distribution networks, software-defined optical networks, and edge computing. Prof. Zhao has hosted and taken part in more than 30 research projects, including the National High Technology Research and Development Program in China ("863" program), the National Basic Research Program of China ("973" program), and National Natural Science Foundation of China (NSFC) projects. He has been granted eight awards from the government and two awards from industry. In 2013, he received the support of Youth Talent Plan Beijing City. In 2018, he received the support of the National Science Fund for Excellent Young Scholars. In 2018, he received the Youth Science and Technology Award of the China Communications Society and the Cooperation Innovation Achievement Award of the China Association of Industry and Research.