AI-Driven Packet Forwarding With Programmable Data Plane: A Survey

Wei Quan, *Senior Member, IEEE*, Ziheng Xu, Mingyuan Liu, Nan Cheng, *Member, IEEE*, Gang Liu, *Student Member, IEEE*, Deyun Gao, *Senior Member, IEEE*, Hongke Zhang, *Fellow, IEEE*, Xuemin Shen, *Fellow, IEEE*, and Weihua Zhuang, *Fellow, IEEE*

***Abstract*—The existing packet forwarding technology cannot meet the increasing requirements of Internet development due to its rigid framework. Application of artificial intelligence (AI) for efficient packet forwarding is gaining more and more interest as a new direction. Recently, the explosive development of programmable data plane (PDP) has provided a potential impetus to packet forwarding driven by AI. Therefore, this paper presents a survey on the recent research in AI-driven packet forwarding with PDP. First, we describe two of the most representative frameworks of the packet forwarding, i.e., the traditional AI-driven forwarding framework and the new one assisted by the PDP. Then, we focus on capacity of the packet forwarding under the two frameworks in four measures: delay, throughput, security, and reliability. For each measure, we organize the content with the evolution from simple packet forwarding, to packet forwarding capacity enhancement with the assistance of AI, to the latest research on AI-driven packet forwarding supported by the PDP. Finally, we identify three directions in the development of AI-driven packet forwarding, and highlight the challenges and issues in future research.**

***Index Terms*—Machine learning, packet forwarding, programmable data plane.**

I. Introduction

PACKET forwarding (PF) is an essential operation of communication in the Internet. Switching devices store and forward received packets through a series of preset processes to complete the delivery of data. However, with the rapid development of network technologies, the exponential growth of global Internet traffic stimulates an unprecedented demand in four aspects: low delay, high throughput, high security, and high reliability [1], [2], [3]. Examples include a line-rate packet forwarding capacitated with 6.50 *Tbps* [4], a low-latency packet forwarding on the order of 0.32 *ms* [5], a secure packet forwarding against high-volume attacks [6], and a reliable packet forwarding for highly dynamic vehicular networks [7]. The traditional framework of packet forwarding is rigid, using the same process in most scenarios. Many efficient algorithms for improving network performance cannot be easily deployed in the traditional framework, such as dynamic network resource allocation for customized network services or attack defending for a secure network.

The software defined network (SDN) architectures decouple network control and forwarding functions, forming a control plane (for control) and a data plane (for forwarding). Various efficient algorithms can be easily deployed on the control plane to control the forwarding logic. In addition, the control plane highly matches the applications of artificial intelligence (AI) in packet forwarding field and brings new possibilities.

AI, as an intelligent algorithm, has great potential for satisfying the aforementioned demands [8], [9], [10]. Compared to other algorithms, the application of AI can flexibly and accurately allocate network resources according to different service demands, enabling the implementation of customized networks. At the same time, the application of AI is real-time. When the network state changes, the resource allocation can be adjusted in time to adapt to network dynamics. For example, AI can effectively model dynamic features of multiple heterogeneous network paths and find the optimal scheme of resource allocation, to maximize the throughput of multi-path packet forwarding [11]. In addition, the application of AI can help improve network security, such as detecting a distributed denial of service (DDoS) attack by identifying complicated packet behaviors [12]. While AI can be applied to effectively realize functions that packet forwarding cannot perform, deploying AI in networks has challenges. For example, deploying AI in the controller in the control plane will introduce unexpected delays, which becomes a stumbling block to high-rate packet forwarding [13], [14]. Furthermore, the effectiveness of an AI model is affected by the data accuracy of network state information. The excellent AI model will also cost many network resources to gather these data.

Manuscript received 11 May 2022; revised 25 September 2022; accepted 11 October 2022. Date of publication 27 October 2022; date of current version 24 February 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1802503; in part by the Natural Science Foundation of Beijing under Grant 4212010; in part by the Beijing Nova Program; and in part by the Major Key Project of Peng Cheng Laboratory under Grant PCL2022Y04. *(Corresponding author: Ziheng Xu.)*

Wei Quan, Ziheng Xu, Mingyuan Liu, and Deyun Gao are with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: weiquan@bjtu.edu.cn; zihengxu@bjtu.edu.cn; mingyuanliu@bjtu.edu.cn; gaody@bjtu.edu.cn).

Nan Cheng is with the Key State Laboratory of ISN, School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: nancheng@xidian.edu.cn).

Gang Liu is with the Department of Fundamental Network Technology, China Telecom Research Institute, Shanghai 200120, China (e-mail: liug8@chinatelecom.cn).

Hongke Zhang is with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China, and also with the PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen 518040, China (e-mail: hkzhang@bjtu.edu.cn).

Xuemin Shen and Weihua Zhuang are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca; wzhuang@uwaterloo.ca).

Digital Object Identifier 10.1109/COMST.2022.3217613

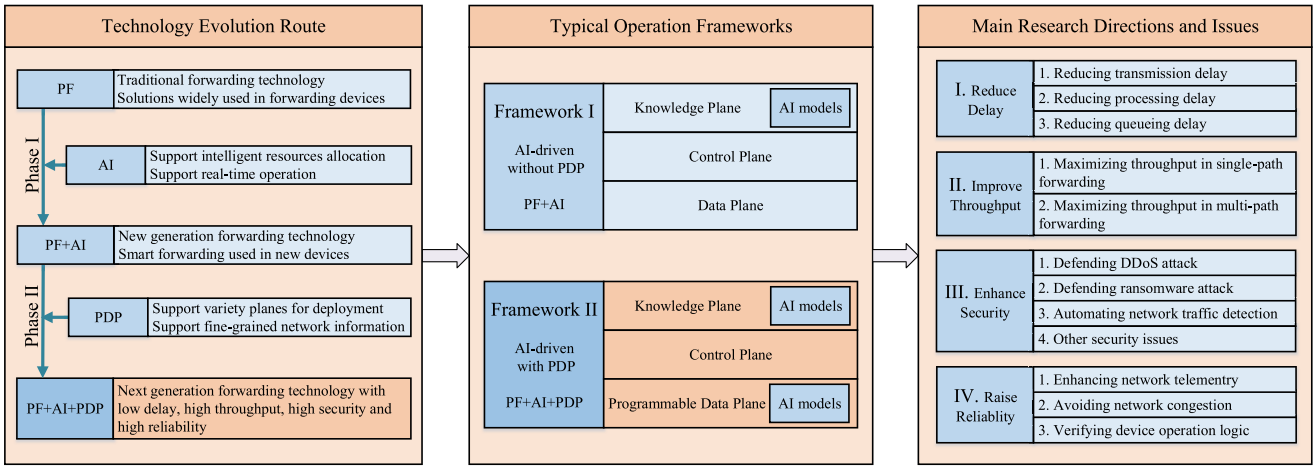


Fig. 1. A Road Map of This Paper.

Recently, the rise of PDP provides a solution to the challenges of AI-based packet forwarding in the SDN architecture, and attracts extensive attention for further performance improvement. On the basis of SDN architecture, PDP enables the data plane programmable and further realizes the customization functions of packet forwarding logic. The PDP provides support for new technologies such as network slicing and priority queueing. The emergence of PDP has also solved the previous problem of AI. The PDP can program the operational process of switching devices so that the algorithm can flexibly choose an appropriate plane to deploy. For example, PDP provides a programmable plane to deply an AI model to avoid the inter-plane delay caused by deploying in a controller [15]. Meanwhile, the PDP can provide detailed network state information for the AI model and only cost a small amount of network resources. Due to the programmability and flexibility, the PDP has a huge potential in AI-driven packet forwarding.

Looking into the evolution of the Internet, we observe that AI and SDN were introduced when traditional PF could not follow the network development, and PDP was created when AI-driven packet forwarding needed further improvement. The evolution of the Internet is accelerated with the convergence of technologies. Globally, there have been extensive research activities on integrating PF, AI and PDP. Existing surveys have summarized research works on AI-driven and PDP-driven packet forwarding separately [16], [17], [18], [19], [20], [21], [22]. Other surveys focus on the integration of AI or PDP with other technologies [8], [23], [24], [25]. A survey is in need to present a broad view of state-of-the-art packet forwarding that integrates the AI and the PDP. Therefore, this paper surveys AI-driven packet forwarding mechanisms and the ones which can be supported by the PDP. To our best knowledge, this survey is one of the first state-of-the-art overviews of the packet forwarding that combines AI and PDP. A road map of this paper is shown in Fig. 1 and the main contribution of this paper is as following:

- First, we provide an overview of the traditional AI-driven packet forwarding framework in SDN architecture and the new framework supported by the PDP.
- Second, we focus on research works that integrate AI and PDP to improve the delay, throughput, security, and reliability performance of packet forwarding based on the two frameworks. Our views of these studies are also discussed.
- Third, we give a brief summary of future research directions and challenges of packet forwarding with AI and PDP.

The remainder of this paper is organized as follows. Section II describes the typical AI-driven packet forwarding framework in SDN architecture and the new framework supported by the PDP. Section III and Section IV discuss research works on improving the delay and throughput performance respectively by the AI and PDP. Then, Section V presents the security performance improvement and Section VI discusses the reliability performance improvement by the AI and PDP. Finally, Section VII identifies three potential directions and open issues in future AI-driven packet forwarding with the PDP and Section VIII draws a conclusion.

All frequently used acronyms in this survey are reported in the Appendix.

II. Frameworks of AI-Driven Packet Forwarding

In this section, we first introduce the framework of AI-driven packet forwarding in SDN architecture, and analyze its advantages and limitations. Then, we give an overview of how PDP can be evolved to enhance the framework (i.e., in overcoming the limitations) and discuss an effective approach that integrates AI and PDP for packet forwarding.

As a new architecture, SDN builds a control plane to decouple the control functions and provides platform for new technologies and efficient algorithms. A traditional SDN packet forwarding framework usually includes a data plane and a control plane. The data plane is the forwarding operation plane. It receives and parses packets, matches the key field with forwarding tables, and sends packets to the next hop. The control plane has a high programmability, which can control the packet forwarding logic in the data plane flexibly to complete some complex functions. For example, the control

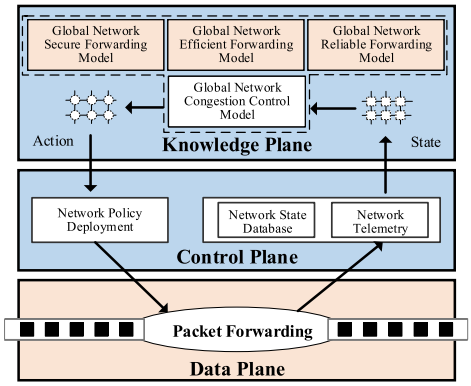


Fig. 2. AI-driven Packet Forwarding Framework without the PDP.

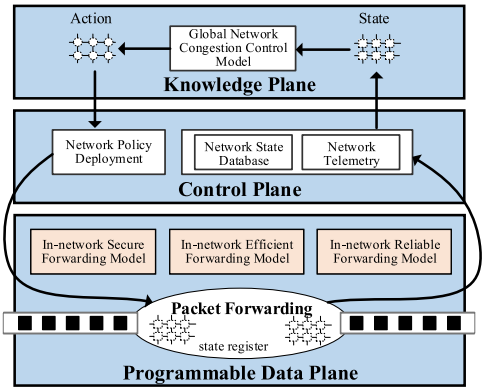


Fig. 3. AI-driven Packet Forwarding Framework with the PDP.

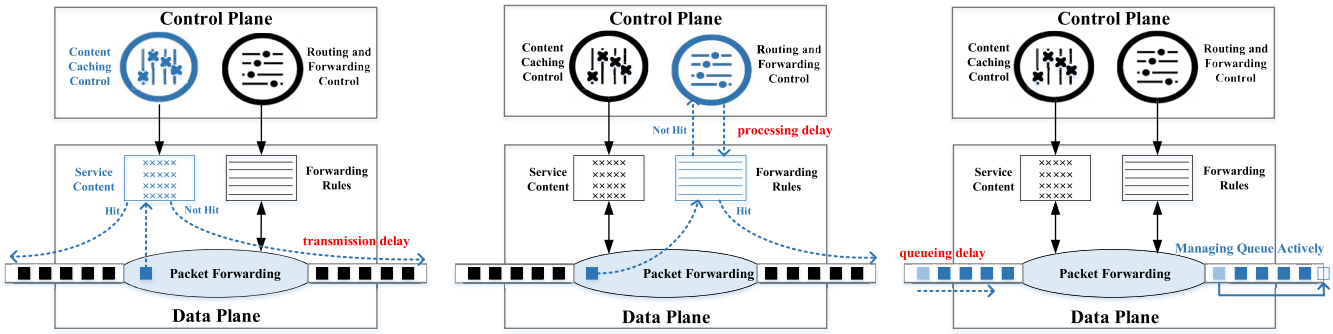
plane can realize the multi-path transmission of packets, as well as the data slicing and congestion control. Furthermore, on the basis of this framework, researchers refine the control plane. They decouple functions such as intelligent calculation and policy translation functions and deploy them in a knowledge plane. The knowledge plane has a high network view and can "observe" the whole network widely and build global efficient algorithms. Therefore, the AI applications are usually deployed in this plane to organize the network information and generate forwarding policies. The functions left in the control plane are responsible for collecting data, translating the new forwarding policies into actions and distributing them to the data plane.

Machine learning (ML), as one of the popular AI technologies, can make packet forwarding more secure, efficient and reliable [8], [26], [27]. As shown in Fig. 2, the AI-driven packet forwarding framework without PDP includes data, control and knowledge planes from bottom to top. Information is transferred from one plane to another (shown by arrows in the figure), which forms a closed control loop. First, the data plane periodically reports network state information (e.g., interface throughput) to the control plane during the process of forwarding packets. Next, the control plane collects and analyzes the information, builds a network state database, and reports the network state to the knowledge plane. Based on the network state, the knowledge plane selects an appropriate forwarding model according to the network performance demand. The forwarding models include global network secure forwarding model, global network reliable forwarding model, and so on. The forwarding model generates corresponding forwarding action policies and the knowledge plane sends them to the control plane. The control plane translates the policies and distributes them through the network policy deployment module. Once the data plane receives the policies, it can adjust the forwarding table and improve the network performance. Finally, a closed control loop is completed and ss the loop continues again and again, the whole network will eventually reach a steady and desired state.

This closed control loop has two drawbacks: (1) *High interaction latency:* The interaction latency between any two planes is long [28], which is not suitable for some delay sensitive applications. For example, the ultra-reliable low-latency communication (URLLC) scenario requires 1*ms* latency [29] which is much shorter than the latency in the AI-driven framework without PDP; (2) *Coarse network state information:* Due to the fixed data plane, network telemetry can only obtain coarse-grained network state information such as interface number and throughput, but cannot obtain fine-grained information such as queue length of interfaces or specific flow rate [30]. Using the coarse-grained information will eventually lead to low accuracy of forwarding models, such as low accuracy in DDoS classification [31].

The PDP provides possibility for overcoming the preceding drawbacks. The AI-driven packet forwarding framework supported by PDP is shown in Fig. 3. Some forwarding models do not need the global network state information and all they need is just the information in one switching device. For example, network security attack identification and interception model only needs the information of received packets. Also, switching device priority queue model only needs the flow type. These models can be deployed on the local device to finish their functions. In comparison with Fig. 2, these kinds of model are transferred from the knowledge plane to the data plane, which reduces the interaction latency. On the other hand, the fine-grained network state information is collected by the PDP. These kinds of model can directly take these information and calculate. Other kinds of models are still placed in the knowledge plane (e.g., global network congestion control model) to complete the overall regulation of the network.

The new framework has two advantages. On one hand, the flexible modification of the packet forwarding process provides a strong foundation for the optimization of operational mode in the AI-driven framework. (1) The PDP can provide wire-speed calculation to reduce ML decision delay (the calculation time of ML model deployed on a programmable switch is about several hundred nanoseconds [32], and the calculation time of ML model on a general platform is tens of microseconds [33]); (2) The match-action pipeline of the PDP can dynamically adapt to the specific network packets such as packets for information collection, and execute their specific forwarding actions. The AI model deployed on the programmable data plane is called "in-network model". Depending on the network requirements, the in-network model has different functions. For example, Xiong et al. propose a mechanism to deploy the in-network traffic classification model on a programmable



(a) transmission delay (b) processing delay (c) queueing delay

Fig. 4. Three Network Delay Components and Corresponding Switch Operating Position (The Blue Part).

data plane to identify and classify service flows [32]. In terms of network security, there are a large number of models, such as in-network abnormal flow monitoring model [34], in-network blackmail monitoring model [35], and in-network DDoS monitoring model [36], [37]. In terms of network performance, there are in-network cache model [38], [39], in-network forwarding rule management model [40], in-network interface queue management model [41], in-network packet scheduling model [42], [43], in-network traffic flow scheduling model [44], [45], in-network small flow scheduling model [46], and in-network domain name system (DNS) model [47]. In terms of network reliability, there are in-network devices running validation models [48].

On the other hand, the PDP can collect the network state information in fine granularity, which facilitates the AI models to realize functions with high performance. For example, Li et al. use the PDP to collect network link state information, and determine whether the corresponding network link is congested [49]. They use an enhancement learning algorithm to minimize the maximum link utilization to avoid network congestion. The proposed global network congestion control model is deployed in the centralized knowledge plane, forming a closed control loop of "data plane uploading information - control plane analyzing information - knowledge plane establishing model and generating policy - control plane translating policy and distributing forwarding action - data plane updating forwarding rules".

The previous two frameworks are the basis of the researches in this paper. In some studies, the knowledge plane is not decoupled from the control plane. Therefore, deploying AI model in the control plane is reasonable. We hope the readers can understand this phenomenon.

III. Network Delay Performance Improvement by AI-Driven Packet Forwarding With PDP

As a low delay is the basic requirement of most network services, the delay performance is usually one of the most important measures to be focused on. This section provides an overview of research works on reducing network delay by exploiting AI-driven packet forwarding with the PDP. The network delay that we discuss in this section is the interaction time from a user sending a request packet to the user receiving a reply packet. In the traditional network, delay can be classified into transmission delay, processing delay and queueing delay. However, in the SDN and PDP architecture, because the switch functions are decoupled, the processing delay is more complex compared to the original one, which not only includes the time of processing packets but also includes the communication time between planes. We still use the name of "processing delay" to facility the reader, but actually the name has a richer connotation. In the following, we discuss the way of the AI and PDP enhance the forwarding function based on the three components of delay.

A. Transmission Delay

The transmission delay represents the transmission time of the packet on the link, which mainly depends on the distance between the user and the data provider, and the selected forwarding path. As shown in Fig. 4(a), when a service request packet of the user arrives at a switch, the switch first matches the service content table. If hit (that is, the service content is cached on the current device), the cached service content will be fed back to the user directly; otherwise, the service request packet will be forwarded to the neighbor switches and the neighbour switches will do the same procedure. The service request packet is continuously forwarded until the service content is found.

When the switch lacks the AI and PDP technology, it cannot easily obtain the link state information of the whole network, so it is difficult to select high-quality paths, such as the shortest path, and will cause long distance between the user and the provider. In order to reduce the distance, some active approaches are adopted, including adding relay nodes or selecting specific relay nodes to cache content, or actively spreading content to the whole network. In the Internet of Things (IoT), it is relatively easy to add virtual nodes in the cloud and fog layer. Therefore, Azath et al. add fog nodes in the network and spread delay-sensitive data between these nodes, to provide the service content to users nearby [50]. At the same time, part of the computing power is deployed on these node to provide users nearby with the computing power service. However, it is difficult to add nodes in mobile opportunistic networks, so Patra et al. establish an optimization model for the selection of nodes for caching

service content [51]. They try to select relay nodes to realize a low average transmission delay from these nodes to every user who requests the content. Furthermore, they propose a fast algorithm with time complexity ($O(n\log n)$) to solve the optimization model. In addition to using relay nodes, actively diffusing data is also a solution. Majeed et al. propose an active service content dissemination mechanism [52]. The server and switching nodes continuously push the critical service contents to their neighbor switches to achieve content caching. This active dissemination mechanism shortens the distance between the user and the switch which stores service content, and reduces the transmission delay.

When the AI is introduced for packet forwarding, the selection of relay nodes becomes more flexible and accurate. The AI applications can analyze users' behavior and predict the contents that users may need. At the same time, The AI applications can control the switches of the whole network through the controller to achieve accurate cache of content. Tsai et al. build a deep learning model for social media networks [53]. They obtain the user's conversation information and adopt the convolutional neural network (CNN) to analyze the context of the sentence, and then predict the service content that the user may request. According to the prediction, the corresponding service content is cached in the switch near the user in advance. The model can reduce the transmission delay by 30% as compared with the traditional method. Compared to the [53], Cheng et al. add individual content request probability (ICRP) to the prediction model to predict the users' request and cache the service content [54]. Then, they use reinforcement learning and Bayesian learning to further improve the accuracy of prediction. Furthermore, the centralized AI model is prone to the disclosure of user privacy information and influences the data security. Saputra et al. design an active caching mechanism for service content caching with distributed deep learning [38]. They collect the user information in several distributed models to protect users' privacy, and further reduce errors in content requirement prediction. In [55], Liu et al. makes a prediction from the other side. Instead of analyzing user needs, they propose a deep-learning-based content popularity prediction (DLCPP) mechanism. The mechanism collects the spatial-temporal joint distribution data of network state and predict the content popularity. The highly popular content is cached in nodes to provide users nearby.

The PDP can only provide limited help in term of transmission delay. It mainly uses its flexible programmability to provide rich and accurate network state information for the AI models and increase the accuracy of model performance.

B. Processing Delay

The processing delay is the most complex part in network delay and contains many component. As shown in Fig. 4(b), when a request packet is forwarded, a switch that receives a request packet will first check if there is any request content stored locally. If not, the request packet needs to be forwarded. The switch matches with the table of local forwarding rules. If hit, request packets will be forwarded according to the rules to the next switch to request content; otherwise, the packet information is uploaded to the controller to request a new forwarding rule. The controller generates and distributes new forwarding rules according to the network state information so that the request packet can be forwarded correctly and the user can quickly acquire corresponding content. In this mode, the size of the storage space in the switch, the load balance of the switch and the controller, and the speed of packet processing in the switch all have important impact on the processing delay. These three parts are the main issues in this section.

In the traditional packet forwarding framework, switches store the forwarding rules. However, the storage space of switch is limited, and too much forwarding rules may cause the overflow. Therefore, it is a feasible solution to store part of the forwarding rules in the control plane. But this solution brings extra delay in the communication between two planes. Therefore, Luo et al. formulate an NP-hard problem to balance the delay of the two planes communication and packet processing [56]. They design a heuristic caching algorithm for forwarding rules to control the interaction between switches and the controller, and minimize the end-to-end transmission delay of data packets. Huang et al. study the compromise point between the cost of switch caching and the cost of network controller distributing forwarding rules, and build a minimum weighted flow provisioning model [57]. According to the network flow information acquisition, they design an off-line algorithm (i.e., after obtaining the network flow information) and two online algorithms (i.e., before getting the flow information) to optimize the deployment of forwarding rules. In addition, before the packet is forwarded, the controller has already initialized forwarding policies to the switch. However, when a large number of users connect to the same node, the switch and the controller will face a huge load, which affects the operation performance. Therefore, Bera et al. design a mobility-aware adaptive flow-rule placement scheme for mobile access networks [58]. In the presence of user mobility, the scheme predicts the next possible location of the user through a K-order Markov chain, and deploys the forwarding rules to the corresponding switches in advance through the active forwarding rule deployment method, so as to reduce the network process delay.

The optimal solution of rule storage strategy calculated by aforementioned heuristic algorithm is static, which is not suitable for the dynamic network. The AI applications can be used to find a better way for rule storage and further improve the network delay performance. Mu et al. design a mechanism based on deep reinforcement learning to analyze and optimize the storage balance between switches and controller, which aims at reducing the controller overhead while ensuring the switch forwarding rules storage [59]. The AI applications can also help with load balancing. Filali et al. present a load balancing mechanism for multiple SDN controllers to reduce the processing time of deploying forwarding rules [40]. First, the mechanism predicts the load of controller through auto regressive integrated moving average (ARIMA) and long short-term memory (LSTM). Then, according to the predicted results, a migration algorithm for forwarding rules based on reinforcement learning is adopted to avoid a large service response

delay in the high-load controller, which can minimize the processing delay of deploying forwarding rules.

The PDP is different from AI in reducing the processing delay. The load balancing problem is a global issue, which cannot be solved directly by the PDP alone. But on the other two issues, the PDP plays an important role. Grigoryan et al. redesign the caching architecture based on the PDP and the field programmable gate array (FPGA) for the massive forwarding rules in the backbone network [60]. They design two caching layers architecture and use FPGA to replace traditional ternary content addressable memory (TCAM) in hardware to accelerate the cache speed and realize fast forwarding table matching when packet is forwarded. Zhang et al. also propose a new data plane architecture [61]. They use behavioral level caching mechanism to cache the packet processing behavior of multiple forwarding tables in a unified manner, so that the switch can still ensure a low processing latency under complex forwarding rules. In contrast to [61], Zhang et al. propose the ShadowFS system with small caches space to store and manage entries. The switch can obtain high frequency information and feed these information back to the controller to adjust the forwarding rules in time.

C. Queueing Delay

Queueing delay is mainly reduced by active queue management (AQM) mechanism in the network. AQM contains a large number of priority queueing algorithms, which can control the delay of flows by adjusting the forwarding order of packets. As shown in Fig. 4(c), in the traditional packet forwarding framework, packets of various service flows enter the queue according to their arrival time at the switch, and then leave the queue according to the first-in first-out (FIFO) principle after processing at the switch. This approach can complete packet forwarding but is not suitable for some delay-sensitive service flows.

Without AI and PDP, the AQM algorithm has been relatively mature in tradition network. Jung et al. design a smart AQM mechanism based on CoDel for the unique network fluctuation of 5G network [62]. The mechanism dynamically adjusts the queue priority according to the queue state, reduces the queueing delay, and can also cope with the network fluctuation. Olariu et al. design a queueing mechanism based on packet delay requirement [63]. The mechanism classify packets in voice over Internet protocol (VoIP) into five priorities according to the delay requirement and put them into different push-in first-out (PIFO) queues. The design can avoid most congestion of data packets and reduce the end-to-end delay. Qiu et al. propose a backpressure queue scheduling algorithm in order to make the switches respond to emergency packets in a timely manner under the elephant flow [64]. The algorithm provides the shortest forwarding path for emergency packets and ensures that queues at the switches will not be congested on this path, so as to reduce the end-to-end network delay.

With the help of AI, the queueing delay can be further reduced. The approaches can be classified into two main ways. On one hand, the AI applications can better classify the service flows and preferentially forward delay-sensitive services. Alnoman design a two-class priority queueing system based on supervised learning [41]. They train and test the system from simulated data sets and then identify delay-sensitive applications in the IoT network based on characteristics such as type and location. The system assigns high priority queueing for the delay-sensitive applications to reduce the end-to-end delay. On the other hand, the AI applications can predict the demand of service flow and forward these flows with different priority. Zhu et al. propose SmartTrans, which provides multi-level priority queues for different flows [65]. They use deep learning to classify and predict the ranking of different flows, and provide corresponding priority queues according to the prediction results. In addition, they expand the buffer of the switch and improve throughput when the network flow surges.

Because of the programmability of the PDP, the data plane is the future platform to deploy AQM algorithm. However, how to implement AQM on the PDP have become new challenges. The researches mainly focus on this issue. Papagianni and Schepper implement an AQM algorithm (PI2) that needs only the information on the data plane [66]. This algorithm uses PDP and the information of inlet and outlet pipeline of the switch. The queueing delay can be reduced in a short time once PI2 is deployed. Kundel et al. also implement an AQM algorithm (CoDel) based on PDP [67]. These researches demonstrate that AQM can be supported for queue management in a network composed of programmable switches. Alcoz et al. propose a solution (strict-priority PIFO (SP-PIFO)) to the challenge of deploying PIFO on PDP hardware [68]. Specifically, they use the P4 language to dynamically adjust the priority of packets based on network state and provide corresponding queues. The SP-PIFO can be fully implemented on Barefoot Tofino switches with a low hardware overhead.

In addition, realizing AI-based AQM is also a challenge. Some AI-based AQMs are so complicated that it is difficult to be deployed in traditional switches. The PDP provide a solution to this issue. Shi et al. design a two level queueing management mechanism based on the AI and PDP [69]. This mechanism uses OpenFlow switch architecture and extreme gradient boosting model (XGBoost) to accurately obtain queueing delay of switches and provide differentiated service priority for applications, which can improve the quality of service and reduce the end-to-end delay. The PDP not only can provide a platform for deploying complex AI-based AQMs, but also can provide refined network state information. Zhang et al. propose an application classification method based on a hybrid deep neural network [70]. This method automatically obtains a large amount of accurate network flow processing information through PDP, and then learns it to classify applications at a high accuracy.

D. Summary and Remarks

We start our review from related techniques about three components of network delay, which is transmission delay, processing delay and queueing delay. Then, we discuss how AI and PDP can help further reduce the three network delay components. In the following, we provide an conclusion of the

three components, their corresponding research and discuss our views.

Transmission Delay: The transmission delay is directly affected by the distance and transmission path between the user and the service content provider. Using relay nodes to cache data is an effective way to shorten the distance. Without the AI and PDP technologies, researchers often use algorithms to calculate the best location of relay node in static network to meet the demand of nearby service providing. However, these algorithms can not adapt to the dynamic change of the network well, resulting in non-optimal results. Actively pushing service content to the network is another solution, but this solution will bring a large amount of data traffic to the network, affecting other network communications.

When AI is introduced, the selection of relay nodes becomes more accurate and reasonable. Most researches determine the relay nodes based on two stages: predicting service content popularity and choosing their storage location.

In the first stage, the research can be classified into the analysis of user behavior and the analysis of service flow. On one hand, by analyzing the user's behavior, the corresponding service content popularity model can be established to predict the service required by the user. On the other hand, services can also be directly observed to analyze their demands, so as to predict the popularity of each service. No matter which analysis perspective, the prediction model requires accurate data. Therefore, the introduction of the PDP can provide guarantee and make the prediction results more convincing.

In the second stage, according to the service content popularity, the content will be cached in relay nodes to provide users with nearby service. The research in this part is relatively simple, but there are many research points. When going into the second stage, selecting an appropriate location should also consider the switch state and link state. For example, if a switch which stores a large amount of high popularity contents is broken, its stored contents cannot be quickly provided to users and the service quality is affected. Alternatively, when the transmission link is congested, a switch closest to the user no longer has the minimal transmission delay. In short, the choice of caching location is not simply a ranking of content popularity, but also depends on the actual network environment. Therefore, further research on AI models is required to consider more about network state. The accurate and rich network information provided by the PDP matches to this future trend.

Processing Delay: The processing delay is mainly affected by three aspects: the storage location of forwarding rules, the load of switches and controllers, and the logic complexity of switches. The first two aspects need to be balanced between switches and controllers, while the third aspect is affected by the operational architecture of the switch. Therefore, AI and PDP have different roles in the studies of reducing processing delay.

Without the AI and PDP technologies, the research mainly relies on heuristic algorithms to balance the performance of the switch and controller. Whether pursuing the minimum delay or the minimum total cost, it is the solution in a static network. But when facing the dynamic network changes, the algorithms are not suitable.

Using AI applications can help the algorithms adapt to dynamic networks. The AI applications can predict the future network state based on the obtained network state information, so that forwarding rules can be classified and deployed in different devices in advance. However, when the network state changes, the forwarding rules need to be adjusted. There are limited studies on the ways of adjusting existing forwarding rules, which need to be researched in the future.

The PDP can reduce the processing delay in two ways. On one hand, the PDP can provide the network state information for the AI model. On the other hand, PDP can adjust switches in the hardware and architecture level, such as increasing the caching space in the switch or using new hardware to increase the logic operation speed. These are the unique capabilities of the PDP which makes it irreplaceable in future packet forwarding.

Queueing Delay: The essence of reducing the queueing delay is how to design the AQM. The traditional AQM has plenty of algorithms for different optimization objectives, such as achieving low delay or high throughput, but these algorithms mainly adjust the packet forwarding order in the queue.

The AI applications can help AQM in another aspect. By predicting the service flow demand, the AI applications can accurately classify the service flows and assign them with different priorities to ensure the high-quality transmission. The programmability of the PDP can provide platform for flexibly deployment and realize various intelligent and complicated AQMs. Also, the PDP can provide accurate network state information for flow classification. However, related research on integration of AI and PDP is still in its infancy. There will be more studies on AQM based on both AI and PDP, as it deserves further research.

The three components of network delay are independent of each other in operation, and their effective integration will lead to a comprehensive delay reduction mechanism. As the Internet requires lower and lower network delays, more and more studies on this topic will continue to appear. The studies discussed in this section for reducing network delay based on AI and PDP are summarized in Table I for easy reference.

IV. Network Throughput Improvement by AI-Driven Packet Forwarding With PDP

The exponential growth of network traffic increases the throughput demand for various services. This section discusses the approaches to improve throughput in traditional network, and presents how to further improve the throughput with the AI and PDP.

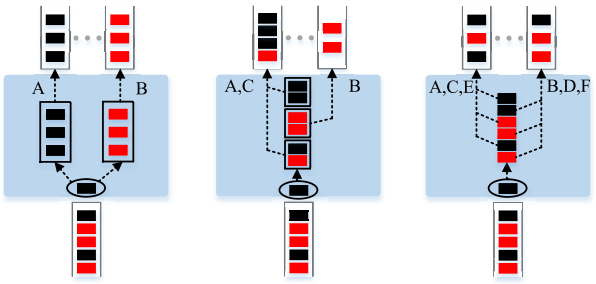
With the popularity and development of the Internet, the amount of packet transmitted in the network increases exponentially, and the throughput performance of various services needs to be further improved. Before we talk about the approaches for throughput improvement, we need to clarify two important concepts: throughput and bandwidth. Throughput and bandwidth are important metrics in network

TABLE I
SUMMARY OF PUBLICATIONS ON REDUCING NETWORK DELAY BASED ON AI AND PDP

Paper	Reducing which component of delay[1]	AI or PDP	Algorithms, AI Models	Network type of application	Main ideas
[53]	1	AI	Deep Learning, Convolutional Neural Network	Mobile social media network	Apply sentence analysis on the data, predict service requirements by deep learning and cache content in advance.
[54]	1	AI	Bayesian Learning, Reinforcement Learning	The wireless network	Predict user preferences and individual content request probability by Bayesian learning, cache content by reinforcement learning in advance.
[38]	1	AI	Distributed Deep Learning	Mobile edge network	Collect user information by content servers, predict users' content requirements by distributed deep learning while ensuring privacy.
[55]	1	AI	Deep Learning	Information-centric networking	Obtain the spatio-temporal joint distribution data of network state, predict the popularity of content by deep learning and cache the service content with high popularity.
[59]	2	AI	Deep Reinforcement Learning	Data center network	Classify forwarding rules by reinforcement learning, deploy the commonly used rules in the switch and the rest in the controller.
[40]	2	AI	Reinforcement Learning Long Short-Term Memory	Multi-access edge computing networks	Predict SDN load and migrate high load in advance to relieve local network pressure.
[61]	2	PDP	Behavioral-Level Caching	-	Unify all cache entries in a switch.
[71]	2	PDP	ShadowFS	Service provider network	Manage entries in small cache and improve the ability to obtain accurate information with high change frequency.
[60]	2	PDP	Programmable FIB Caching Architecture	Backbone network	Redesign cache architecture, replaces TCMA to FPGA and accelerate caching speed.
[41]	3	AI	Supervised Learning	IoT network	Distinguish application priorities by supervised learning and provide two-class priority queues.
[65]	3	AI	Deep Learning	Data center network	Distinguish the priorities of service flows by deep learning, set up multiple priority queues, expand the switch caching space.
[66]	3	PDP	PI2	-	Implement PI2 by the inlet and outlet pipe information of switch on the PDP.
[67]	3	PDP	CoDel	-	Implement CoDel on the PDP and prove that PDP support AQM.
[68]	3	PDP	Push-in First-out Algorithm	-	Dynamically adjust the priority of packets based on network state and provide priority queue.
[69]	3	AI&PDP	eXtreme Gradient Boosting Model	IoT network	Establish two-layer queue management and provide differentiated service guarantee mechanisms for applications.
[70]	3	AI&PDP	Hybrid Deep Neural Network	-	Obtain accurate network information, automatically extract characteristics and classify service flows.

[1] 1: transmission delay; 2: processing delay; 3: queueing delay

performance evaluation. Bandwidth is the theoretical upper limit speed of data transmission, while throughput is the actual value of data transmission speed. Therefore, improving the throughput is to purse the upper limit speed and it has two approaches. One approach is to improve the utilization of bandwidth through algorithms in every single path, so that the throughput can be increased to shorten the distance to the bandwidth value. Another approach is to improve the throughput by increasing the bandwidth. This approach can be realized by multi-path transmission technology. Without the help of the AI and PDP, traditional networks lack methods in the former approach, and mainly rely on the latter one to improve throughput. Specifically, in the latter approach, the packet forwarding mechanism distributes a certain number of packets to different network links according to a scheduling algorithm, so as to achieve link bandwidth aggregation and



(a) service flow (b) small flow (c) packet

Fig. 5. Three Flow Granularities of Multi-path Forwarding.

maximize network throughput. According to the number of packets to be scheduled (i.e., the granularity of flow), the packet forwarding algorithm can be classified into three categories: (1) multi-path scheduling with service granularity (all packets of a service flow are distributed in a single path); (2) multi-path scheduling with small flow granularity (a fraction of packets of a service flow is distributed in a single path); (3) multi-path scheduling with packet granularity (every packet of a service flow is distribute in a single path) [72].

In multi-path scheduling with service flow granularity, as shown in Fig. 5(a), the packet forwarding mechanism allocates service flows to different network links according to the scheduling algorithm, where all packets in a service flow are forwarded into the same network link. Many algorithms based on service flow granularity are studied. Zhou et al. propose a weighted cost multipath algorithm (WCMP) to control the traffic size of each path [73]. The algorithm sets weights for each path according to the network state information, and allocates service flows in different size into different path, to improve the network throughput. Kheirkhah et al. design maximum multi-path TCP (MMPTCP) algorithm to allocate elephant flows (long duration flows) and mouse flows (short duration flows) [44]. The algorithm solves the problem of poor performance of mouse flows in traditional multi-path TCP (MPTCP) by randomly scattering packets in the network. Wang et al. and Liu et al. use software-defined networking to perceive network topology and link resource information, and adaptively allocate different service flows to network links according to network resource state [74], [75].

The preceding multi-path scheduling algorithms of service flow granularity improve network throughput via matching service flows with different network links. One idea is to allocate the service flows with their unique requirements, such as assigning delay-sensitive service flows to links with a high-quality forwarding environment. Another idea is to allocate the flows based on their sizes. For example, we can distribute an elephant flow on a link with more available resources and distribute a mouse flow on a link with less resources. However, the packet forwarding mechanism based on service flow granularity leads to the different size of service flow in each link, making the utilization of links in an unbalanced state.

In order to avoid uneven utilization of network link resources in the multi-path scheduling with service flow granularity, multi-path scheduling with packet granularity has been studied to distribute packets in a service flow to different network links, as shown in Fig. 5(c). All the red rectangles represent packets belonging to the same service flow and are assigned to two different network links. Zhang et al. propose Hermes, a network balancer, to allocate link resources according to packet granularity [76]. The Hermes can ensure high utilization of network link resources by sensing the network state of available link resources and cautiously adjust the forwarding rules. Dan et al. utilize the centralized control characteristic of the SDN to obtain the network link state information, and present a multi-channel scheduling mechanism with packet granularity to improve the network resource utilization [42]. Liu et al. propose a scheduling method of packet granularity on the basis of PDP, encapsulate different network-layer headers for different packets of one service flow, and forward them through different network links [77], [78]. The method not only improves the secure transmission capability of packets, but also achieves network link bandwidth aggregation. The preceding packet granularity multi-path forwarding algorithms improve network throughput by fitting different network links with different packets. However, due to unique characteristics of different network links such as in terms of delay, the packet granularity multi-path forwarding algorithms suffer from packets out of order [79].

In order to overcome the problems of uneven utilization of network link resources with the service flow granularity and packets out of order with packet granularity scheduling, a compromise is made with small-flow granularity scheduling which breaks up the service flows into a number of small streams and distributes the streams to different network links, as shown in Fig. 5(b). Xue et al. design a small-flow multi-path forwarding mechanism, dynamically scheduling and adjusting packet with feedback, which enables service flows to be flexibly "cut" and distributed to network links [80]. The mechanism ensures that packets arrive in order and improves the network throughput. Shi et al. develop a multi-path scheduling mechanism based on network state information, which allocates network link resources with small flow granularity and dynamically adjusts the number of packets in a small stream according to network link state [46]. Katta et al. use PDP for network telemetry to perceive resource utilization of network links [81]. They "slice" the service flows into streams according to the gap between adjacent packets arriving at the switch, and distribute the streams to each network link according to the perceived utilization.

When the AI applications are introduced in packet forwarding, it is possible to realize efficient transmission algorithms for improving bandwidth utilization. Different from the bandwidth aggregation method of multi-path forwarding, these algorithms mainly rely on machine learning to complete the selection of data transmission path and distinguish service priority. By assigning different services with different paths, the service throughput is improved. Pasca et al. apply machine learning techniques to the SDN architecture [82]. They establish a decision tree classifier through supervised learning, which can classify and assign priority to service flows according to their protocol and source/destination address. As the switch processes the packets, the switch allocates

different flows to different paths based on the priority. This mechanism improves performance of high-priority services and maximizes network throughput. Azzouni et al. propose NeuRoute, an ML algorithm based dynamic routing framework [83]. The framework uses deep learning to learn service flow features and predict network flow changes. The new forwarding rules can be generated and distributed to each switch, which dynamically change the transmission path to increase network throughput. Mohammed et al. develop a deep reinforcement learning algorithm [84]. By analyzing the state of the wide area network, the algorithm adjusts the transmission path of the service flow, and reconfigures the traffic of each link, which improves the link utilization and the throughput.

The AI applications also play an important role in multi-path transmission. In the multi-path transmission, multi-path scheduling and balancing is the focus of recent research. In order to maximize throughput, it is a normal idea to transmit a small amount of traffic on a small bandwidth link and to transmit a large amount of traffic on a large bandwidth one. The AI applications can help for these flow allocation and scheduling. Ji et al. propose an MPTCP-based automatic learning selection path mechanism (ALPS-MPTCP) [85]. This mechanism automatically obtains the network link states and adaptively selects high-quality paths to transmit data packets. Li et al. design a multi-path packet forwarding congestion control mechanism based on ML to solve the low throughput problem caused by heterogeneous links in multi-path forwarding [11]. They use reinforcement learning to analyze network congestion and dynamically adjust the congestion window, in order to improve throughput aggregation. It is worth noting that the training of reinforcement learning model is an off-line process, which does not influence the decision making process or introduce extra delay and overhead. Gilad et al. present a high-performance multi-path congestion control architecture [86]. This structure uses on-line convex optimization to solve the balance problem of fairness and high performance. Kanagarathinam et al. propose smart multi-path switch (SMS) for MPTCP in a wireless network [87]. The SMS can use machine learning to dynamically adjust and manage MPTCP streams based on the state of the wireless network to improve network throughput.

In the AI-based packet forwarding framework, the integration of the PDP has a relatively simple effect on throughput improvement, which is mainly to provide platform for deploying high-performance network state telemetry algorithm to obtain rich and accurate information. PINT is a typical efficient network telemetry algorithm. By modifying the communication packet and adding a 1-bit field, the network state information can be obtained under normal communication [88].

In the research for bandwidth utilization improvement, Li et al. propose a mechanism to classify application flows with different QoS requirements based on C4.5 decision tree and adjust switch packet queue depth [89]. The mechanism provides different transmission parameters for application flows with different priorities, which can reduce delay and increase throughput for the application flows with the highest priority. Liu designs a deep reinforcement learning based routing algorithm [90]. The algorithm transforms the performance requirements of service flow into the resource requirements. Then, according to the requirements, the algorithm classifies service flows and allocates corresponding resources. In addition, the algorithm pays attention to network states regularly, adjusts forwarding rules adaptively, and optimizes network resource allocation.

In the research for multi-path transmission, Liu utilize PDP to measure fine-grained network state information and adopt a deep Q-learning model to make decisions on packet forwarding path selection [91]. The model can reduce the out-of-order packets rate in parallel multi-channel transmission, which improves the network throughput. Hardegen and Rieger implement feature prediction for network flow based on ML and collect a variety of heterogeneous network link state information based on PDP [92]. They select each packet forwarding path according to the feature prediction and network link states to maximize the network throughput. Hu et al. propose EARS, an intelligence-driven experiential network architecture for automatic routing [93]. The EARS uses deep reinforcement learning to control switches and modify forward policies by interacting with the network environment to improve network throughput.

Summary and Remarks: In the traditional Internet, throughput is mainly improved by multi-path forwarding. Packet forwarding algorithms with three flow granularities have their own advantages and limitations. Algorithms based on service flow can effectively transmit the same service flow over a unique link, but will lead to uneven utilization of link bandwidth. Although algorithms based on packet granularity can improve link bandwidth utilization, but will lead to disordered packets. The forwarding algorithms based on small flow is between the two. It can achieve an appropriate utilization of bandwidth while decreasing out-of-order packets. It achieves satisfactory performance in all aspects, but not outstanding. On the other hand, forwarding based on small flow is a static balance. When the network fluctuates, the balance is broken, and the throughput performance deteriorates significantly.

With the introduction of the AI and PDP, there are more ways to improve throughput, which makes it possible for improving bandwidth utilization. Both in the two improvement approaches (single-path forwarding and multi-path forwarding), the AI applications can dynamically select network links through various models to achieve reasonable resource allocation and purse high throughput. In addition, in the multi-path forwarding, the AI applications can change inter-path balance from static to dynamic, and can adapt to various network fluctuations within a certain range, and stably maintain a high performance of network throughput for a long time. Related researches on the PDP are relatively simple, which only provide rich network state information for the AI models. The scalability and the flexible programming plane of the PDP do not show much in terms of throughput improvement. The research on the AI and PDP for throughput performance improvement is still at an early stage. In the future, studies on more AI algorithms and PDP utilization method will be carried out, to further enhance throughput improvement.

TABLE II
SUMMARY OF PUBLICATIONS ON IMPROVING NETWORK THROUGHPUT BASED ON AI AND PDP

Paper	AI or PDP	Algorithms, AI Models	Network type of application	Main ideas
[82]	AI	Supervised Learning	-	Classify and prioritize service flows, assign different flows to different paths based on priority.
[83]	AI	Deep reinforcement learning	Wide area network	Predict network traffic changes and generate new forwarding rules to improve network throughput.
[85]	AI	K-NN Algorithm, Random Forest, K-Means Algorithm, Reinforcement Learning	IoT network	Automatically obtain network link states and adaptively select high-quality paths to transmit packets.
[11]	AI	Reinforcement Learning, Hierarchical Tile Coding Algorithm	Heterogeneous network	Dynamically adjust the congestion window size to reduce congestion and improve aggregation throughput.
[86]	AI	Online Convex Optimization	-	Use the online convex optimization to solve the fairness between paths in the actual network environment and achieve high performance transmission.
[87]	AI	Smart Multipath Switch	Wireless network	Dynamically adjust and manage MPTCP substreams according to the state of the wireless network.
[88]	AI&PDP	In-band Network Telemetry	-	Add 1 bit in packet to obtain information such as delay, queue depth, and link utilization to achieve fine multi-path control.
[89]	AI&PDP	C4.5 Decision tree	-	Classify applications and configure priorities for services, adjust queueing depth of SDN switches to forward packets with different priorities.
[90]	AI&PDP	Deep Q-Network, Deep Deterministic Policy Gradient	-	Transform performance requirements of the service flow into the resource requirements, allocate network resources for flows and adaptively adjust the forwarding rules.
[91]	AI&PDP	Distributed Asynchrono Deep Reinforcement Learning	-	Analyze the network states and determine the forwarding path, reduce the packet out-of-order rate.
[92]	AI&PDP	Deep Neural Network	-	Predict network traffic in every path and select an appropriate one to forward.
[93]	AI&PDP	Deep Reinforcement Learning	-	Interactive network environment periodically and modified forwarding rules dynamically.

The research works on improving network throughput performance based on AI and PDP discussed in this section are summarized in Table II for easy reference.

V. SECURITY PERFORMANCE IMPROVEMENT BY AI-DRIVEN PACKET FORWARDING WITH PDP

Ensuring data security of terminals and forwarding devices is gaining increasing importance, which puts forward a high requirement on the network security performance. This section summarizes recent works on how the AI and PDP can improve packet forwarding in terms of security. That is, AI-driven packet forwarding with PDP can defense network attacks, such as distributed denial of service (DDoS), ransomware, abnormal network traffic detection (ANTD), and other security issues, with high accuracy and low latency, and can mitigate the attacks with high efficiency.

There are various kinds of network security issues, and we select several typical network security attacks in this section. First, we discussed DDoS and ransomware attacks. In the network, DDoS attacks tend to target servers or important nodes. An attacker controls multiple systems and constantly sends malicious traffic to suppress the server, host or application, which makes the computing or networking system overload. As a result, the important node breaks down and cannot provide services properly. A DDoS attack can be regarded as a server-side attack. however, ransomware is on the opposite side. Ransomware attacks user's device by infusing viruses. It hijacks important data and extorts money from the user. So, ransomware is a user-side attack. Defense against user-side attacks is different from defense against server-side ones. The detail will be discussed in the following sections. Then, we talk about the ANTD. ANTD no longer focuses on specific types of network attacks, but analyzes whether the network traffic itself is abnormal. Therefore, the ANTD is more difficult compared to the DDoS and ransomware, because traffic anomalies contain a large number of types. This makes the identification and classification of traffic challenging. Finally, we briefly discuss some other network security issues, including link flooding attack (LFA) and eavesdropping.

A. Distributed Denial of Service

Distributed denial of service (DDoS) attack is that an attacker controls multiple systems and constantly sends

malicious traffic to suppress the server, host or application, which makes the computing or networking system overload. In the defense of traditional DDoS attack, a switch identifies the fields of forwarding packet, and filters out the DDoS attack packets by extracting feature information. However, due to the limited performance of the traditional switch, the defense method cannot reach a high speed and high efficiency.

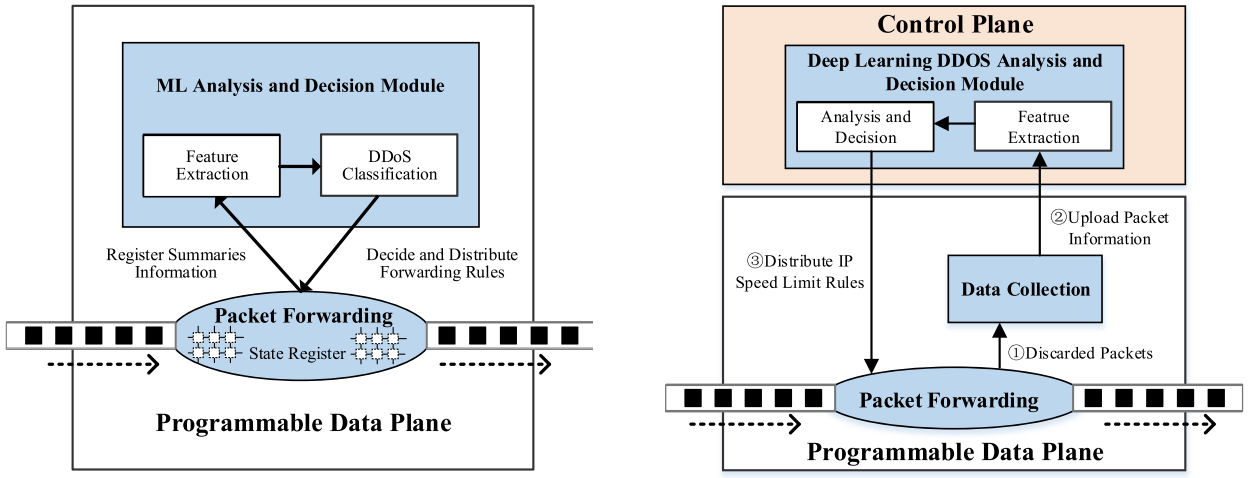
As the network evolves to the SDN architecture, the target of DDoS attack expands. The controller in SDN architecture, as the core of the whole network, undertakes most of the network work. Controllers have become the new target for DDoS attacks. An attacker impinges massive traffic to the controller and paralyzes it, which causes the switch network to fail to work properly. In order to solve this problem, researchers studied the DDoS attack defense for the SDN controller. Shang et al. propose FloodDefender, a network defense framework, to address the problem that communication links between two planes in the SDN architecture are vulnerable to DoS attacks [94]. The framework monitors network states in real time. When an attack occurs, the neighbor switch takes over the work of the victim switch and sends the received data stream to the interceptor of the controller to identify whether it is DoS attack. In addition, the controller updates the new attack stream features in the interceptor in real time to improve the efficiency of interception. Chen et al. use extreme gradient boosting (XGBoost) as a detection method for cloud-based SDN networks [95]. The XGBoost classifier detects DDoS attacks on packets collected by TcpDump. The method has high detection accuracy, low false positive rate, fast detection speed and scalability.

Servers are also vulnerable to DDoS attacks in the network. With the development of technology, DDoS attacks can be better disguised as normal packets, which challenging the identification and classification algorithms in switches. The AI applications can deal with this issue due to its excellent learning ability. In the complex network environment, the AI applications can learn the features of known DDoS attack packets and detect new DDoS attacks accurately and automatically. In the IoT network, attackers use insecure consumer grade devices to carry out DDoS attacks on critical infrastructure. Doshi et al. propose a mechanism that uses IoT network specific behavior, such as a limited number of IoT nodes, regular forwarding time interval between adjacent packets, to conduct DDoS attack feature analysis [12]. In particular, they first capture network traffic, sorting packets based on information such as IP address and time. Then, they extract features from the packets and classify them into normal traffic and DDoS attack traffic.

In the Internet, DDoS attacks are widely studied. Labeling the data set of DDoS attack to complete the supervised learning is the first defense method. Idhammad et al. design a DDoS detection mechanism based on semi-supervised learning to solve the problems of low accuracy and high false positive rate in typical DDoS detection [36]. The detection mechanism consists of two main parts: unsupervised learning and supervised learning. Unsupervised learning estimates the entropy of network traffic features based on a time sliding window algorithm, and then calculates the information ratio of gains. Network traffic with high information ratio of gains is considered abnormal. Supervised learning uses an classifier based on an extra-tree algorithm to accurately classify abnormal traffic and reduce the false positive rate of unsupervised learning. The detection failure rate of the mechanism is low. The second defense method is deep learning. Yuan et al. propose a DDoS attack detection method, DeepDefense, based on deep learning [96]. The DeepDefense uses a bi-directional recurrent neural network and data sets to learn patterns from network traffic sequences and track network attack activities. The Deepdefense is better than shallow machine learning method in recognition error rate and generalization. Niyaz et al. propose a multi-vector DDoS detection system based on deep learning [97]. The system monitors network traffic, analyzes and evaluates traffic tracks in different scenarios, and determines whether DDoS attacks exist. Using support vector machine (SVM) to classify data traffic is also a defense method. Hu et al. propose a prototype system to defend DDoS attack in real time [98]. During the process of packet forwarding, they adopt flow-based information collection methods in switches to quickly gather network state information, and use the SVM model for data analysis to detect and judge DDoS attacks. At the same time, they design a DDoS attack mitigation mechanism based on the white list, which can dynamic update the packet forwarding rules to block the service packets that are not in the white list and effectively reduce the damage of DDoS attacks. Furthermore, the combination of SVM and self organizing map (SOM) can better defend against DDoS attacks. Phan et al. design a hybrid stream working mechanism based on SVM and SOM [99]. The mechanism uses SDN to collect switch flow information and classify the flow by SVM. Then, SOM is used to make the final decision, and the switch transmission rules can be changed to reduce network attacks. To compare the performance of these AI algorithms, Santos et al. realize three different types of DDoS attack detection (flow table attack, bandwidth attack and controller attack) using four machine learning methods: SVM, multilayer perceptron (MLP), decision tree and random forest in an SDN simulation environment [100]. The results show that the decision tree approach has the shortest processing time, and the random forest approach has the highest absolute value accuracy.

In SDN architecture, the AI algorithms are often deployed in the controller. As a result, the communication time between the data plane and the control plane makes the real-time effect of the model decrease and affects filtering DDoS attack. The PDP can directly deploy the defending algorithm on the data plane. Prez-Daz et al. design an PDP-based security architecture consisting of two independent systems, an intrusion prevention system (IPS) and an intrusion detection system (IDS) [101]. The IDS is deployed at the host, and is trained by machine learning for network traffic identification. If a flow is recognized as an attack based on information collected by the PDP, the flow will be processed by the IPS model to defend against the DDoS attackers. The proposed architecture is practical for identifying and mitigating DDoS attacks.

The integration of the AI and PDP is the development direction of DDoS attack defense, which is mainly studied



(a) Detection Framework Based on Forwared Packets

(b) Detection Framework Based on Discarded Packets

Fig. 6. Two Mechanisms about Defending DDoS Attacks with AI and PDP.

in three aspects. Firstly, the PDP provides multiple planes to better deploy AI models, which can realize a DDoS monitoring algorithm on wire speed and greatly reduce monitoring latency [102], [103]. Secondly, fine-grained network state information can be supplied by the PDP, including queue length, network delay and so on, to provide more multidimensional features for accurately identifying DDoS attacks [104], [105]. Thirdly, the flexible and programmable characteristics of the PDP provide the foundation for customized DDoS attack monitoring [106]. The DDoS attacks monitoring algorithm can be dynamically selected according to requirements of security level. Therefore, aggregating the AI and PDP for defending DDoS attacks has become a research hotspot in the community.

Musumeci et al. use the AI and PDP to detect DDoS attacks during packet forwarding in real time [107]. Their mechanism workflow is shown in Fig. 6(a). In the packet forwarding process with the PDP, the numbers of IP, UDP, TCP and SYN packets are counted, and the local ML analysis and decision module can use these numbers to judge whether there is a DDoS attack. If there is, the decision forwarding engine distributes the forwarding rules containing the attacker's IP and other information in the form of table entry, and blocks the attacks. Mi and Wang design an ML-Pushback mechanism with deep learning and PDP to accurately identify and mitigate DDoS attacks [108]. The workflow is shown in Fig. 6(b). The data collector is customized on the PDP, and gathers real-time discarded packets in the network. When the number of discarded packets reaches a threshold, the data collector automatically extract packet information and uploads them to the control plane. When the deep learning module of the control plane receives the information, the feature extraction sub-module will extract the IP source/destination address, protocol type, interface number and other features, and then hand them to the decision tree of the analysis and decision module to judge whether there is a DDoS attack. If a DDoS attack exists, the features of DDoS attacker are further extracted. Then, the analysis and decision model distributes the forwarding rules to switches to limit the traffic from the attackers. Chen et al. view the network from another aspect. They propose a distributed intrusion prevention system, CIPA, for programmable networks based on artificial neural network (ANN) [109]. The CIPA distributes computing power to some or all of the switches in the network, monitors the network and forms a global view of the network states, and performs high-precision intrusion detection and mitigation on discovered malicious online traffic.

B. Ransomware

Ransomware is a kind of virus software that extorts the files of victims. The attacker holds the file until the victim pays enough money to redeem it. Because ransomware viruses exist on user devices, studies in this area tend to focus on traffic in the user side.

In traditional networks, defenders identify ransomware by analyzing the data flow of user devices. The introduction of the AI applications does not fundamentally change this detection logic, only increases the speed and efficiency of identification. The AI algorithms have different way to defense ransomware. The first way is based on the data information on the user's device. Poudyal et al. analyze the raw data, library files, number of functional interface calls and other multi-level data of the user's computer, and use Bayesian network, random forest and other supervised learning algorithms to detect ransomware [110]. Also, ransomware can be found by analyzing user opcodes. Zhang et al. first analyze the opcodes of the user terminal and generate an N-gram sequence, then extract the features from the sequence with the TF-IDF data mining algorithm, and finally construct a ransomware detection model with the extracted features [111]. The third way does not target the raw data, but identifying ransomware by analyzing the information entropy of the data. Lee et al. make comprehensive use of information entropy and AI methods to classify ransomware [35]. Their classification scheme can accurately identify ransomware with low

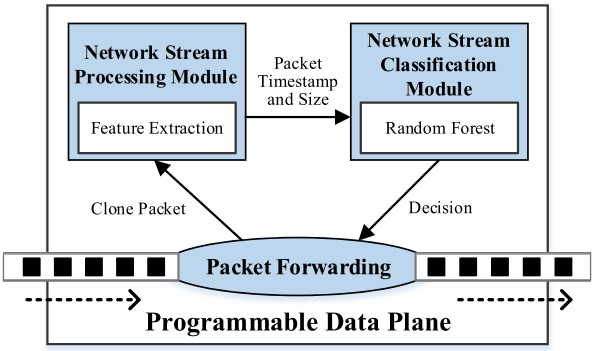


Fig. 7. Mechanism on Defending Ransomware Attack with ML and PDP.

false positive and false negative rates compared with the existing detection methods. Analyzing network interface traffic is the forth way of detecting ransomware. Alhawi et al. design NetConverse to identify ransomware [112]. Specifically, they analyze network flow, extract its features, and compare them with the features of the learned ransomware attacks to determine whether it is a ransomware flow. The last way is to build a firewall to block ransomware. Shaukat and Ribeiro propose RansomWall, a layered defense system for defending against crypto-ransomware [113]. RansomWall combines static and dynamic analysis to extract file features and send them to the feature collector. The feature collector sends suspicious file to the machine learning model to make the final judgment on whether it is a ransomware software.

Although there are many ways to defend against ransomware, AI-based defense modes are on the user side. These defense modes are passive and can only be triggered when the attack reaches the user. However, if the PDP technology is introduced into ransomware defense, the passive defense mode can be changed into an active one. The new deployment location of defending algorithms is on the network side, which can identify and detect ransomware attacks early. Once the attack traffic enters the network, it can be monitored, identified, and intercepted by the PDP switch, so that the attack traffic cannot reach the user's device. Cusack et al. propose a ransomware precise detection mechanism based on the AI and PDP [114]. As shown in Fig. 7, the mechanism mainly includes two modules: network stream processing module and network stream classification module. First, the network stream processing module uses the PDP to quickly collect the information of each packet, and extracts the network stream features (packet 5-tuple information), including packet timestamp, packet size and byte number. Then, the network stream classification module gathers the feature information to identify the ransomware stream by the random forest algorithm. By adjusting the number (40) and depth (15) and the maximum amount of features in random forest decision trees, the accuracy of detection ransomware reaches 86%, and the failure rate of detection is only 11%.

C. Automating Network Traffic Detection

Automating network traffic detection (ANTD) mainly addresses the attacks caused by abnormal network traffic. Especially in the data center, a large amount of network traffic needs to be monitored for a safe network environment. However, the ANTD technique has low performance in a traditional network and brings potential risks for network security.

Similar to the detection of DDoS attacks, analyzing and identifying abnormal data traffic through the AI applications has great advantages in network security. The AI applications can construct an abnormal network traffic model and accurately detect abnormal traffic, which achieves a high security level. Due to its wide range, the ANTD has been studied in various networks, and the parameters used in the AI models are different. Salman et al. establish an abnormal traffic detection framework for the IoT network based on ML [34]. In this framework, network traffic information is collected from the IoT network devices. Then, the information is extracted into 39 network traffic features for further classification. The ML algorithm keeps running to identify abnormal network traffic with a high accuracy. However, in the P2P networks, dynamic port numbers make it difficult to analyze abnormal traffic based on former AI method. Additional parameters is required to identify abnormal flows. Kong et al. design an anomaly encrypted traffic detection method based on machine learning and behavior features [115]. This method extracts behaviour features of application programs and classifies abnormal traffic based on machine learning. This method can effectively improve the accuracy of abnormal encrypted traffic detection. In big data networks, the detection performance of unbalanced abnormal traffic needs to be further improved. Yong et al. design a parallel cross convolutional neural network (PCCN) for network traffic detection [116]. The PCCN fuses two branches of convolutional neural network with excellent learning ability. It can complete the learning of traffic characteristics in the case of a small number of samples. In addition, it adopts an improved original flow feature extraction method and achieves fast convergence speed of network traffic classification and identification, which lead to a quick detection.

In the Internet, ANTD has the most related researches, most of which use the SVM to identify and classify traffic. Boero et al. use the SVM algorithm as the core of the system to detect abnormal traffic with the features of byte rate, packet rate and average packet length [117]. Their scheme has a high detection rate. Bhunia and Gurusamy establish an SDN-based framework called SoftThings, where switches continuously monitor the traffic of IoT devices and provide the traffic information to the cluster SDN controller [118]. The controller detects abnormal behaviors by analyzing traffic, and dynamically sets traffic rules in the switches. In this process, abnormal traffic can be quickly detected at the network edge. Kong et al. propose an abnormal traffic identification system (ATIS) based on the SVM [119]. The system determines abnormal traffic in four steps: data collection, traffic features extraction, data processing, and SVM classification. The system can classify and identify various attack traffic applications.

Random forest, deep learning and other algorithms can also be used in ANTD. Song et al. propose an abnormal traffic defense mechanism based on SDN and machine

learning [120]. The mechanism first extracts the features of network traffic, then evaluates the malicious degree by using a random forest algorithm, and finally makes defense decisions to effectively prevent abnormal traffic transmission. Ajaeiya et al. design a lightweight traffic detection and defense system [121]. The system periodically collects traffic information from the SDN OpenFlow switch. Then, it extracts and aggregates a set of features to analyze traffic information, so as to achieve high performance abnormal flow detection. Tuan et al. use deep learning to detect abnormal traffic in an SDN environment [122]. They use the NSL-KDD data set to build a deep neural network anomaly flow detection model. Then the model is used to analyze network traffic and identify abnormal traffic. Experiments show that the model has great potential in abnormal flow detection.

In addition to the above studies, there are also related research on AI model optimization. Ji et al. utilize ML to model a deep attack pattern of network abnormal traffic [123]. Specifically, they design a mechanism based on ML and SVM algorithm to analyze abnormal network traffic and predict the categories of network attacks. Niu et al. propose a network intrusion detection method based on transfer component analysis (TCA) [124]. The method uses TCA to map the traffic features of the source domain and the traffic features of the target task to a shared subspace for domain adaptation, and then uses K-nearest neighbors (KNN), SVMs and random forests (RF) as base classifiers for training detection models to find abnormal traffic.

The high processing performance of the PDP facilitates the wire-speed analysis and detection of network abnormal traffic. The PDP can be used to design a fine-grained traffic collection and analysis mechanism in the packet level to ensure real-time and accurate abnormal traffic analysis [125], [126], [127], [128]. Using PDP to monitor network traffic can avoid sending too many network packets to the centralized control plane, and can reduce abnormal network traffic analysis decision delay [129]. Therefore, integrating the AI and PDP to improve ANTD has become a hot topic in the security field.

Lee and Singh propose an endogenous abnormal traffic analysis mechanism based on PDP, and deploy a random forest algorithm on switches to realize real-time online abnormal traffic detection in networks [130]. The specific workflow is shown in Fig. 8. First, 12 network traffic features are selected as the judgment basis, e.g., the TTL from source to destination terminal (STTL), the TTL from destination to source terminal (DTTL) and the number of packets from destination to source terminal (DPKTs). Then, the 3-order random forest decision tree algorithm is used to carry out abnormal analysis of network traffic. If abnormal network traffic is confirmed, the corresponding packet forwarding rules will be distributed to the switches and the abnormal traffic packets are discarded; otherwise, the packets will be forwarded following the default flow table rules. Busse-Grawitz et al. deploy a supervised learning model on the top plane of PDP to detect the endogenous traffic in the network [15]. First, due to the memory and floating operation constraints in the PDP, they modify the feature selection process in the supervised learning to adapt the information from the PDP. Then, they design a real-time

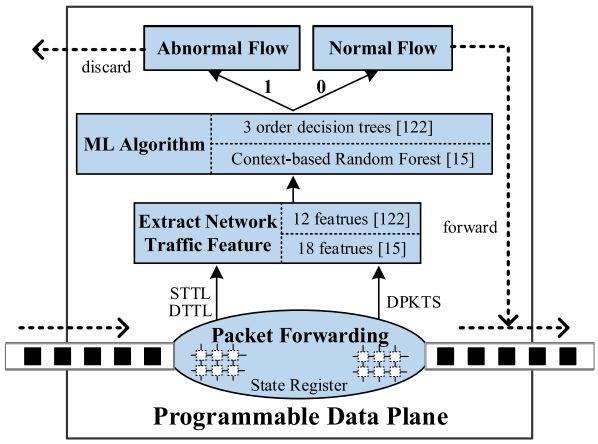


Fig. 8. Two Mechanisms about ANTD with AI and PDP.

accurate network traffic analysis mechanism based on the AI and PDP. In the model training stage of the mechanism, the labeled network traffic data is used to train the classification model of supervised learning, and then the classification model is deployed on the PDP to realize the wire-speed online classifier. The workflow is shown in Fig. 8. First, 18 network traffic features are extracted by counting the first three packets of a network traffic as the judgment basis. Then, the random forest algorithm with semantic association is used to conduct abnormal network traffic analysis. If abnormal network traffic exists, the corresponding traffic will be discarded according to the packet forwarding rules; otherwise, the packets will be forwarded following the default flow table rules.

D. Other Security Issues

In this section, we discuss some other network security issues and their defense methods, mainly including link flooding attack (LFA) and eavesdropping.

LFA is a variant of DDoS attack. Different from traditional DDoS attacks, attacker for LFA sends a large amount of normal traffic to cause network congestion in network links, and finally isolates the target area from the Internet [131]. Attack traffic in LFA is normal traffic, so it is difficult to identify LFA as DDoS attack, which puts forward new requirements for network security defense. In particular, in SDN architecture, because of the centralized characteristics, attacking SDN controller can isolate the switch network controlled by it and paralyze the network device on a large scale. Traditional feature extraction, classification, and filtering of traffic cannot effectively defend against this attack. Therefore, gaining other information of traffic is helpful in this issue. Rasool et al. propose CyberPulse [132]. They use the deep learning model to continuously monitor the communication traffic in SDN architecture. In addition to extracting content features, they also analyze the behavior of the traffic and judge whether it is an LFA.

The eavesdropping attack occurs inside the link. The attacker can analyze the data content by eavesdropping the communication traffic, resulting in data leakage. In addition to keeping the data confidential to protect the content,

the defense party can also proactively intercept eavesdroppers by analyzing and identifying eavesdropping behaviors. Hoang et al. classify the traffic behavior by one-class SVM in the unmanned aerial vehicle network, and then use the K-means algorithm to cluster the classified behavior and judge the eavesdropping attack [133]. Xiao et al. propose a beamforming control scheme for visible light communication based on deep reinforcement learning, which intelligently adjusts the beam structure according to the behavior of eavesdroppers to enhance information security [134]. Liu et al., combine the AI and PDP, and propose a three-layer defense mechanism for eavesdropping, including (1) identifying abnormal network states and adjusting forwarding decisions through the minimum risk machine learning algorithm, (2) splitting sensitive data through multi-path transmission, and (3) highly compatible with the original defense means [135].

E. Summary and Remarks

Section V describes various types of network security issues and corresponding defense methods. These network security attacks threat the network security from different sides. The main target of DDoS attacks is the server. By sending malicious packet to the server, the attacker will eventually make the server overload and stop working. Ransomware attacks users by hijacking their important data through viruses in order to extort money. ANTD targets at identifying a broad range of abnormal data traffic. LFA attacks the network domain and isolates the specific network by causing link congestion. Eavesdropping targets data packets and analyzes important information about users by stealing the contents of data packets. Different network security issues have different defense methods, and we will briefly summarize these methods and discuss our views.

DDoS: DDoS attacks exist on all types of networks. The main defense method is to analyze the features of data traffic. So, how to identify DDoS attack packets more efficiently and accurately is the research direction. The AI applications can learn the features of known DDoS attack packets, identify and filter the new type attack packets. While providing accurate packet information, the PDP can support an ML model to run on the data plane, avoiding the process of data upload and policy distribution, and improving the real-time interception. Existing solutions have the capability for successfully filtering DDoS attacks with high accuracy. However, identifying and filtering functions are in general deployed in the switch near the victim, which is relatively passive and only triggered at the arrival of DDoS attack packets. Is it possible to intercept DDoS attacks at the edge of the network when the packets enter the network? If so, it will not only successfully defend against DDoS attacks early, but also reduce a large number of junk packets in the network.

Ransomware: Ransomware attacks user devices through viruses. In the early days of the network, ransomware is defended primarily through user-side analysis. The defending methods analyze user's information to detect ransomware, including data, operation behavior, information entropy and so on. Although it can effectively defend against ransomware attack, on the one hand, it is triggered once the attack reaches the user's device, which makes the defense passive. On the other hand, it may cause privacy issues due to the analysis of user information. Therefore, in the recent research, the deployment location of defense method is changed from the user side to the network side. With the help of two powerful technologies, the AI and PDP, ransomware attacks can be detected and intercepted sooner and faster.

It is worth noting that the ransomware blocking rate of each study is generally less than 90% which is not sufficiently high as desired. The reason is that ransomware can be packaged in various types of packets, which is very difficult to identify. In the future Internet, there will be more protocol format of packet, which will lead to higher level for ransomware identification. Identifying them and extracting common features of ransomware packets correctly and quickly is the future research direction. How to integrate the AI and PDP to maximize ransomware defense capabilities requires further studies.

ANTD: ANTD has defensive effect against all kinds of abnormal network traffic. However, because it does not focus on the defense of specific security issues, the ANTD is extremely difficult and challenging. ANTD defense model requires a large amount of network state information and complex feature extraction and identification algorithms. So, the technology integration effect of the ANTD, AI and PDP is the best and has the most related research. The development direction of ANTD is not further integrating technology, but reducing computational load and increasing accuracy. How to optimize AI model and how to acquire targeted network information is the research direction of ANTD.

LFA and Eavesdropping: LFA, as a variant DDoS attack, presents a great challenge for network security. The data traffic of LFA is normal traffic, and traditional feature extraction cannot effectively identify this attack. Although data traffic behavior analysis can increase the accuracy, it still overlooks a large number of LFA. Therefore, how to better identify LFA is the future direction. The security problem of eavesdropping is not closely related to packet forwarding technology, and it is mainly defended by data encryption. However, packet forwarding technology can defending eavesdropping in other ways. For example, multi-path transmission is a good direction. By splitting the data and transmitting it over different paths, the network can defend single-way eavesdropping. At present, there are few researches on using packet forwarding to solve eavesdropping issue, and we hope there will be more novel perspectives in the future.

The papers related to network security improvement based on the AI and PDP discussed in this section are summarized in Table III, Table IV and Table V respectively for easy reference.

VI. Reliability Performance Improvement by AI-Driven Packet Forwarding With PDP

Reliability refers to the quality of data transmission and is an important issue of the network. Reliability is mainly affected by two aspects: (1) Congestion of transmission link;

TABLE III
SUMMARY OF PUBLICATIONS ON DEFENDING DDOS ATTACK BASED ON AI AND PDP

Paper	AI or PDP	Security type	Algorithms, AI models	Main idea
[12]	AI	DDoS	Random Forests, SVM, K-nearest Neighbors, Decision Trees, Neural Networks	Classify packets, extract traffic characteristics, and classify them into normal traffic and attack traffic.
[33]	AI	DDoS	Semi-supervised Learning, Time Sliding Window Algorithm, Extra-trees Algorithm	Unsupervised learning extracts the characteristic entropy of data stream and calculates the information gain ratio. Supervised learning distinguishes abnormal flow and reduces the false positive rate of unsupervised learning.
[96]	AI	DDoS	Bi-directional Recurrent Neural Network	Use bi-directional recurrent neural network and data sets to learn DDoS attack patterns and track attack activity.
[97]	AI	DDoS	Deep Learning	Monitor network traffic, analyze and evaluates traffic tracks in different scenarios, and determine whether DDoS attacks exist.
[98]	AI	DDoS	sFlow-based Method, Supervised Learning, Support Vector Machines	Use sflow-based information collection to quickly collect network state information, use support vector machines to analyze and identify DDoS attacks, update forwarding rules in time, and minimize looses caused by DDoS attacks.
[99]	AI	DDoS	Support Vector Machine, Self Organizing Map	SDN collects switch information. Support vector machine classifies data flows. Self organizing map determines whether data flows are attack flows and modifies switch forwarding rules to reduce attacks.
[100]	AI	DDoS	Support Vector Machines, Multiple Layer Perceptron, Decision Tree, Random Forest	Test the effectiveness of different machine learning algorithms against DDoS attacks.
[101]	PDP	DDoS	Intrusion Prevention System, Intrusion Detection System	Intrusion prevention system uses machine learning on the host to identify DDoS attacks. DDoS attacks are sent to the intrusion detection system of the controller for processing and interception.
[107]	AI&PDP	DDoS	Random Forest, Knearest Neighbors, Support Vector Machine	Collect the number of IP, UDP, TCP and SYN packets, analyze the data and identify DDoS attacks by local ML, update the flow rules to intercept DDoS packets.
[108]	AI&PDP	DDoS	Deep Learning, Decision Trees	Collector collects the packets discarded by the switch and sends the packets to the controller for feature extraction. The controller identifies DDoS attacks and updates the forwarding rules in the switch to limit the rate of DDoS attacks.
[109]	AI&PDP	DDoS	Artificial Neural Network	Assign computing tasks to multiple switches, collect data from each switch, and form network state monitoring to detect DDoS attack flows with high accuracy.

(2) Operation logic of the forwarding device. The first aspect mainly corresponds to the network congestion control mechanism, while the second aspect mainly corresponds to the network forwarding device self-verification mechanism. These two aspects will be discussed in detail in the following sections. In addition, the connection of the network is complex, and it is difficult to find network congestion or faulty device. So, researchers often use network telemetry technology to find problem links or devices. Among these network telemetry algorithms, in-band telemetry (INT) has become the most commonly used technology because of its simplicity and quick detection. Therefore, network telemetry will be discussed first.

A. Network Telemetry

Network telemetry refers to how information from various data sources is collected by using a set of automated communication processes and transmitted to the corresponding device for analysis tasks. INT, as the most commonly used technology for network telemetry, is deployed in the PDP. By extending the fields of the packet, telemetry data can be stored and transmitted to the collector along with the packets. However, the expansion of packet fields inevitably brings extra overhead for transmission. Therefore, the expansion mode of packet fields has become one of the key points of research.

Basat et al. propose a network state telemetry mechanism based on PDP to collect information such as interface queue length, packet forwarding processing delay, packet forwarding interface [88]. The mechanism needs to insert only one bit telemetry information when packets are forwarded. The network state telemetry information can be partitioned and restored in multiple packets, which greatly reduces the overhead of telemetry. Janakaraj et al. design a distributed in-band network telemetry system (S-INT) and a wireless network operating system (WINOS) [136]. The S-INT reduces the overhead of monitoring network traffic by embedding a specialized INT header in the data stream. At the same time, the WINOS system can collect distributed telemetry information and connect with an SDN network to realize fast machine learning algorithm and network control. Vestin et al. develop a fast INT reporting collector based on the PDP [137]. The collector uses P4 language and is deployed on the stream processor of switches. It can quickly obtain the switch data plane information, while requiring a low network overhead.

TABLE IV
SUMMARY OF PUBLICATIONS ON DEFENDING RANSOMWARE BASED ON AI AND PDP

Paper	AI or PDP	Security type	Algorithms, AI Models	Main idea
[110]	AI	Ransomware	Bayesian Network, Random Forest, Supervised Learning	Analyze the original data, library file, number of function interface call of user, and detect ransomware by supervised learning algorithm.
[111]	AI	Ransomware	Term Frequency-Inverse Document Frequency Data Mining Algorithm	Analyze the operation code of user and generate N-gram, extract sequence features by term frequency-inverse document frequency data mining algorithm, construct ransomware detection model.
[35]	AI	Ransomware	Information Entropy, Decision Tree, Deep Learning	Classify ransomware according to different file formats based on information entropy and ML methods.
[112]	AI	Ransomware	Decision Tree	Analyze network flow, extract features, compare with ransomware feature database, determine whether it is ransomware.
[113]	AI	Ransomware	Gradient Tree Boosting Algorithm	Extract file features by static and dynamic analysis, send features to collectors. Collector send suspicious files to machine learning engines to identify ransomware.
[114]	AI&PDP	Ransomware	Random Forest, Decision Trees	PDP quickly collects data flow information, extracts features. ML collects features and determines whether it is ransomware.

The INT technology can be used to monitor the network state in real time and locate faulty device. Zhou et al. design a real-time traffic monitor [138]. The monitor closely deploys multiple information collection modules in programmable switches and periodically collects switch information to obtain the network states. Holterbach et al. use the PDP to quickly detect network failures, and send rerouting signals to the control plane to ensure rapid recovery of services [139].

When network telemetry is assisted by the AI algorithm, the telemetry performance is further improved. First, the INT technology requires the switch to store the forwarding rules of the INT packets, which poses a new challenge to the switch with limited storage space. Lazaris and Prasanna propose a DeepFlow framework to enable fine-grained measurements in programmable switches [140]. The framework can adaptively measure the activity of service flow and provide different fine-grained network state information for different activity levels. In addition, the framework uses historical measurements to train a cloud-based deep learning model to provide short-term traffic predictions when switch resources are limited. Second, different networks have different telemetry information, and the INT assisted by AI can intelligently select telemetry data. Hohemberger et al. formulate an INT's assignment plan model and enhance it with machine learning algorithms [141]. The enhanced model can improve the ability of INT to identify abnormal network states. Third, the AI applications can deploy INT algorithm in advance and speed up network telemetry rates. Yang et al. propose a high-speed network telemetry mechanism called Fast-INT [142]. The Fast-INT monitors network change events through a reinforcement learning algorithm, and dynamically deploys and adjusts INT monitoring tasks to achieve efficient network monitoring in a short time.

In addition, the AI applications can obtain information from INT to perform high-quality network functions. Mayer et al. propose a soft-failure localization framework based on ML [143]. The framework can be applied in the case of failure of telemetry equipment that cannot use INT to detect. In this framework, an artificial neural network is used to simulate network failures, and the approximate location of failures can be obtained quickly, which speeds up failure location determination. Pashamokhtari establishes a security monitoring mechanism for the IoT network [144]. The PDP is used to dynamically monitor the packet information of various IoT devices. A set of AI models are used to classify and detect the information to find whether there is malicious behavior.

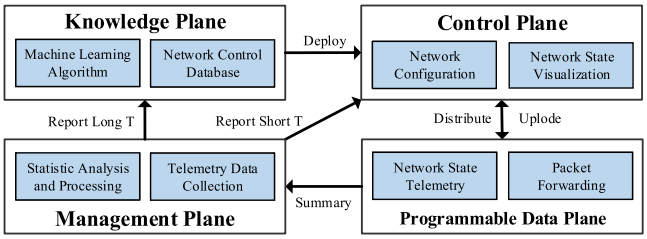


Fig. 9. Framework of Network Telemetry with AI and PDP.

Overall, the combination of AI and PDP has a huge advantage in terms of INT technology. The framework of its operation is shown in [145]. Hyun et al. use an in-band telemetry mechanism of the PDP to obtain accurate network state information. Based on the information, they use ML to build a knowledge-defined network whose system framework is shown in Fig. 9. The system is mainly divided into four planes, i.e., programmable data plane, management plane, control plane and knowledge plane. In the programmable data plane, the network state information is collected and summarized in the process of packet forwarding and uploaded to the management plane and the control plane. The management plane acquires the network state information and then carries on data statistics, analysis and processing, and finally constructs the network state database. The control plane gathers the short-term reports from the management plane and

TABLE V
SUMMARY OF PUBLICATIONS ON AUTOMATING NETWORK TRAFFIC DETECTION BASED ON AI AND PDP

Paper	AI or PDP	Security type	Algorithms, AI Models	Main idea
[34]	AI	ANTD[1]	Convolutional Neural Network, Residual Neural Network, Recurrent Neural Network	Collect traffic information of the IoT device, extract characteristics, and use machine learning algorithms to classify and identify abnormal traffic.
[115]	AI	ANTD	Intrusion Detection Systems, Decision Trees	Extract behavior characteristics of applications, use machine learning to classify and determine abnormal traffic, improve the accuracy of abnormal flow detection.
[116]	AI	ANTD	Convolutional Neural Network	Fuse two branch convolutional neural networks, adopt improved method for extracting original stream features, detect abnormal flow fast and efficiently.
[123]	AI	ANTD	Support Vector Machines	Use support vector machines algorithm to analyze network abnormal flow and predict the mechanism of network attack category.
[117]	AI	ANTD	Support Vector Machines	Collect byte rate, packet rate and average packet length, and detect abnormal traffic by support vector machines.
[118]	AI	ANTD	Support Vector Machines	Collect traffic characteristics, provide them to the controller, analyze and distinguish data flows, and adjust forwarding rules dynamically.
[119]	AI	ANTD	Support Vector Machine	Collect traffic information, extract traffic features, use support vector machine to determine abnormal traffic.
[120]	AI	ANTD	Random Forest	Extract the characteristics of traffic, evaluate the malicious degree, and make decisions to prevent malicious flow transmission.
[121]	AI	ANTD	Bagged Trees	Periodically collect traffic statistics, extract and aggregate features to analyze traffic information, and implement high-performance abnormal traffic detection.
[122]	AI	ANTD	Deep Learning	Establish anomaly flow detection model based on NSL-KDD data set, detect the abnormal flow.
[124]	AI	ANTD	Transfer Component Analysis, K-Nearest Neighbors, Support Vector Machine, Random Forests	Extract traffic characteristics of source and destination domain, carry out domain self-adaptation, classify and determine the abnormal traffic.
[130]	AI&PDP	ANTD	Supervised Learning, Random Forest, Decision Tree	Collect 12 network traffic characteristics, use decision tree to analyze whether the network traffic is abnormal.
[15]	AI&PDP	ANTD	Supervised learning, Random Forest	PDP rapidly extracts 18 traffic features. Supervised learning realizes abnormal traffic identification.

[1] 1. Automating Network Traffic Detection

the information from the programmable data plane, and then show the network configuration and state for visualization. The knowledge plane acquires the long-term data of network state from the management plane, extracts the features of the network state and trains the ML model to form the knowledge of network traffic scheduling and abnormal detection. Then, the packet forwarding policies are generated through the knowledge and deployed to the control plane. The control plane translates the policies into packet forwarding rules and distributes the rules to the programmable data plane.

B. Network Congestion Control and Management

Network congestion control and management is to ensure the network transmission performance remains at a high level for a long time. The traditional congestion control mechanism is to adjust the congestion window to reduce the degree of congestion when it occurs, which is very passive. With the help of AI, the passive approach can be changed into active one. By establishing a congestion prediction model, the AI applications can detect network congestion in advance and adjust forwarding rules in time to avoid congestion. Jain et al. obtain a large amount of data from practical telecommunication network and train the network traffic prediction model [146]. The model can predict the occurrence of network congestion and use big data to adjust the packet forwarding rules and improve the quality of service. Zhou et al. propose a supervised learning regression model that can capture global routing behavior [147]. The model can obtain whole network routing information and predict network congestion.

The PDP enhances the congestion control mechanism from the source of network state information. Feldmann et al. use PDP to identify elephant flow at wire-speed level and assign an independent queue for each elephant flow through multi-queue management [148]. Further, they monitor queue state information in real time to detect network congestion. If congestion is about to occur, a congestion signal will be sent to inform the upstream node to change the packet forwarding rules, which can avoid network congestion and achieve accurate congestion control.

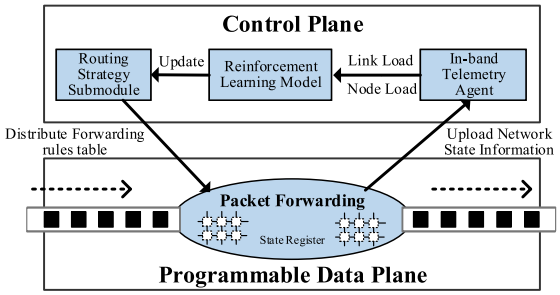


Fig. 10. Framework of Network Congestion Control with AI and PDP.

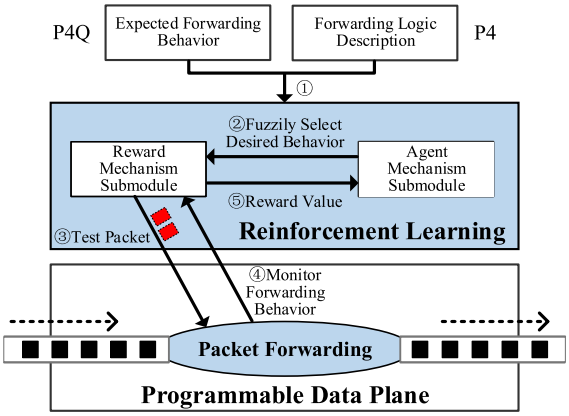


Fig. 11. Framework of Runtime Verification with AI and PDP.

In recent years, researchers have explored using the AI and PDP to realize efficient and predictable in-network congestion management. Mai et al. propose a network congestion detection and management mechanism based on reinforcement learning for network congestion caused by instantaneous burst of elephant flows [149]. When an elephant flow fluctuates, the mechanism can adjust queueing assignment based on real-time feedback from reinforcement learning to avoid network congestion. Li et al. design a TCP-proximal policy congestion control (TCP-PPCC) algorithm [150]. The algorithm obtains network state information through the PDP, updates the forwarding policy offline, and further adjusts the new policy online. It can effectively prevent network congestion.

The typical congestion control framework combining the AI and PDP is shown in [49]. Li et al. use the PDP to accurately collect network link state information, based on which, they determine whether the corresponding network link is about to be congested. Further, they adopt an enhancement learning algorithm to minimize the maximum link utilization to avoid network congestion. The workflow of this mechanism is shown in Fig. 10. First, the network state information is collected through the INT when the packets are forwarded in the PDP. Then, the INT agent in the control plane gathers the collected network state information and extracts the load of each network link and each node device as the state parameters for reinforcement learning. After that, the reinforcement leaning model generates routing adjustment policies to avoid network congestion based on the state parameters. Finally, the control plane translates the policies into packet forwarding rules and distributes them to the network devices on the programmable data plane.

C. Network Device Runtime Verification and Management

With the popularity of programmable switches, customized packet forwarding becomes the basis of more and more research works. The smooth and complete forwarding logic can reduce the interruption of packet transmission and greatly improve the packet transmission performance. At the same time, refined forwarding code can reduce useless processing logic and improve data processing performance. Along with the trend, verification and management of the functional feasibility of programmable network devices and the validity of the runtime forwarding table have become a research hotspot. The functional feasibility verification of network device refers to the analysis of whether the PDP packet forwarding processing has bugs [151], [152], [153], [154]. The validity of the runtime forwarding table refers to the analysis of whether there is an abnormal packet forwarding rules in matching tables when the PDP is running [48], [154], [155]. For example, malicious attackers can tamper a forwarding table to interrupt the user's services [154]. A verification mechanism is completed by Zhou et al. They design an anomalous runtime forwarding table mechanism [48]. The mechanism uses an intermediate representation based on a binary decision diagram to form the data plane probe of switch and to monitor the anomaly of packet forwarding. The mechanism can ensure that packets are transmitted efficiently and smoothly across the PDP.

In addition to manually collecting network packet forwarding state information to verify and manage the functional feasibility and the validity, researchers explore the AI applications to realize automatic verification and improve the correctness and feasibility of packets forwarding management. Jagadeesan and Mendiratta formulate an approach to enhance automated verification with machine learning-based analysis and detect the faulty or malicious behaviors [156]. The approach takes the switch behavior as input and determines whether there is an faulty or malicious behavior through machine learning analysis. Furthermore, the approach has low operational requirements and can easily be deployed on the switch. Shukla et al. propose a variation-coefficient and fuzzy-evaluation mechanism guided by reinforcement learning to verify the validity of packet forwarding logic and forwarding table during the operation of programmable network devices [157]. The workflow is shown in Fig. 11. First, the users describe expected packet forwarding behavior on the PDP through P4Q lightweight language, and provide packet forwarding logic described by P4 language. Second, the reward system of reinforcement learning generates two kinds of test packets according to information from the users. One is the ordinary packet, which is used for samples (seeds) of the initial environment state of reinforcement learning. The other is specific boundary packets (such as Ethernet address FF:FF:FF:FF:FF:FF), which is used to verify whether there are bugs in the packet forwarding logic. The reward system sends the ordinary test packets to complete the initialization of the reinforcement learning system. Then, the reward system

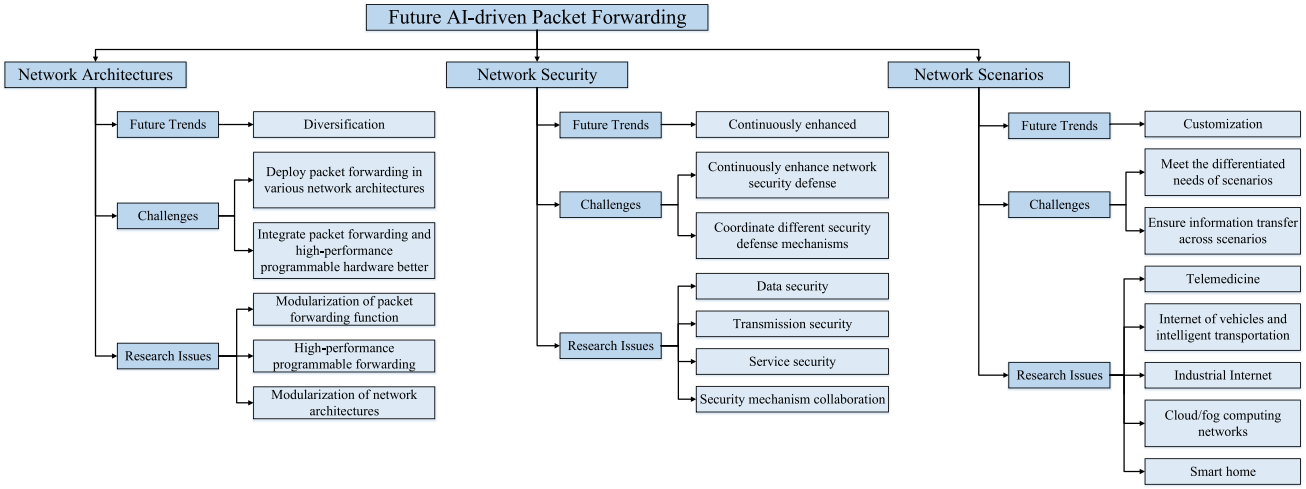


Fig. 12. Future Trends, Challenges and Open Issues of AI-driven Packet Forwarding.

sends out a specific boundary test packet, at which time the agent fuzzily selects an action expected to forward the packet. Meanwhile, the reward system monitors the forwarding process of the packet in real time. Finally, the forwarding of this packet is finished and the reward system compares the two actions and gives a reward value. The agent checks whether the reward value triggers the bug threshold. If it is, the agent will notify the user that there is a bug in the packet forwarding.

VII. Further Trends, Challenges and Open Issues

Packet forwarding is the foundation and the core of the Internet and still has a long way to go in the future. AI-driven packet forwarding technology is the basic development direction, and intelligence is one of the most important features of future networks [158], [159]. This section discusses the future AI-driven packet forwarding technology from three development trends, and highlights challenges and research issues respectively. The structure of this section is shown in Fig. 12.

A. AI-Driven Packet Forwarding for Diverse Network Architectures

Future Internet architecture is one of the most discussed research hotspots in recent years. Traditional IPv4 networks can no longer meet the needs of the current diversified and intelligent network requirements. Thus various new network architectures have emerged, such as open programmable network (ForCES, SDN), new service-oriented network system (SOI, NetServ, COMBO, SONA), content center network (NDN, DONA, PSIRP, NetInf), new mobility-oriented network system (MobilityFirst, HIP, LIN6, Six/One), intent-driven networks (IDN), and smart identifier networks (SINET). The explosion of proposed new network architectures has brought about today's Internet, where the traditional IPv4 network is deployed as the core and diversified new network archictures are accessed as private networks.

1) Challenges: Packet forwarding technology is an indispensable part in both traditional and new network architectures. However, different network architectures have different deployment planes and functional requirements for packet forwarding, which leads to the first challenge of future packet forwarding technology – *How to properly deploy packet forwarding in various network architectures?* In addition, to satisfy differentiated network architecture requirements, high-performance programmable hardware appears and can be utilized for efficient and customized packet forwarding. *How can we better integrate packet forwarding and high-performance programmable hardware?* The integration becomes a new challenge. On one hand, diverse network architectures have different requirements, and the compatibility of high-performance programmable hardware in the context becomes a problem. On the other hand, the enrichment of forwarding functions requires high-performance hardware to provide more programming interfaces, which undoubtedly puts forward new challenges for hardware design.

Furthermore, a careful look at today's Internet results in a question: whether today's Internet with IPv4 as the core network and new architectures as the private networks will continue to have a development momentum? The new architecture is supposed to break the bottleneck of the traditional Internet and replace IPv4 network. But the actual situation is that the widespread IPv4 provides poor compatibility between new network architectures. New network architectures can exist only in the form of private networks. *Can we develop a super network?* We call this super network as "unified network". It should have excellent forwarding compatibility, which can replace IPv4 network completely at low cost. At the same time, it should have extensive backward compatibility, which can easily access various new networks and truly realize the integration of the Internet.

2) Research Issues: In response to the challenges, we identify three research issues.

- *Modularization of packet forwarding functions:* Functional modularity is not a new concept in programmable network. In network architecture ONOS, various network functions exist in the form of APP,

and the ONOS can arbitrarily take part of functions to meet user needs. In essence, packet forwarding is a network function, and its basic elements are the packet storage and packet forwarding. Hence, it is possible to represent it as separate modules with multiple API interfaces and can be embedded in different network architectures. Even if the network architecture changes, packet forwarding can be deployed and updated.

- *High-performance programmable forwarding:* The enhancement of high-performance programmable forwarding can be carried out from two aspects: general hardware chip and interface enrichment. The high-performance programmable forwarding must be deployed on specific network devices. For example, DPDK is deployed on supported NICs and FPGA is deployed on programmable network cards. All these hardware requires unique interface and hardware of forwarding devices. General hardware chips may be an answer. The packet forwarding can be integrated into a chip placed in the general network hardware. Devices in various network architectures can install this network hardware to implement the packet forwarding. Interface enrichment addresses the integration of programmable forwarding capabilities with high-performance hardware. The richer the forwarding functions are, the more interfaces the hardware needs to provide. How to design high-performance programmable hardware supporting multiple forwarding functions deserves further studying.
- *Modularization of Internet architectures:* The modularization of Internet architectures is an important approach towards the super network. Compared to the modularization of forwarding functions, network architecture modularization looks at the network from a higher perspective. The core of the super network can be similar to that of an Android system. The system itself does not realize any network functions, but establishes a framework for the operation of all network functions and coordinates mutual communications of various networks. Such approach allows for freedom to design a unique network architecture and install it in the system as an APP. The network architecture can operate internally as an independent entity while communicating with other network architectures as a subnet.

B. AI-Driven Packet Forwarding for Enhanced Network Security

We discuss network security alone because network security is receiving more and more attentions in the global community. With the rapid advances of information technology, the digitalization of information dominates the industry, and network security has become one of the most important technical issues.

1) Challenges: Network security defense and security attack are an opposing and unified topic. For any security attack, given a sufficient time, each security defense mechanism can be developed. In turn, each security defense mechanism faces vulnerabilities to new security attacks. Therefore, security attack and security defense are two sides of the coin. There is no security attack that cannot be solved and no security defense that cannot be broken. *How to continuously enhance network security defense?* This is a challenge to the development of network security solutions. In addition, most existing network security defense mechanisms defend against a specific type of security attacks. However, paying too much attention to each specific type of security attack defense leads to lack of the coordination between security defense mechanisms and causes new vulnerabilities. *How to defend more types of security attacks? How to coordinate different security defense mechanisms?* These challenging question requires more research efforts.

2) Research Issues: In view of the challenges, we focus on typical network security defense mechanisms related to packet forwarding technology and divide them into four categories: data security, transmission security, service security and collaboration of security defense mechanism.

- *Data securtiy:* Data security includes data encryption and data recovery. For data encryption, data sources or switching devices encrypt important data before sending or forwarding them into the network. In this way, attackers cannot decrypt the packets to obtain the original information. The AI applications can help the switch select an appropriate encryption algorithm and generate random encryption keys to further improve data security. Block-chain establishes an information chain with multiple nodes responsible for security. The information in block-chain is difficult to be modified. Future research may focus on the integration of block-chain and AI-driven packet forwarding technology to realize decentralized data communication and intelligent packet forwarding [160]. In addition, block-graph brings more possibilities for block-chain [161]. Block-graph transforms two-dimensional lines into three-dimensional graphs, further expanding the secure responsibility of nodes and enhancing mutual supervision between nodes.
  In packet transmission, attackers can destroy a packet to make data incomplete when reaching the destination. Data recovery is necessary for that. Data recovery algorithms can use network coding to increase redundancy in a packet. When the packets arrive at the destination terminal, the terminal only needs partial data to restore the complete original information. The network coding and AI-driven forwarding technology can be integrated. A network coding algorithm should be flexibly selected by intelligent forwarding technology, so as to ensure more secure data.
- *Transmission security:* If we compare data security to putting data into a safe, then transmission security is protecting the delivery of the safe. One popular approach in transmission security is multi-path transmission technology, which arranges data packets and sends them into different paths to hide the transmission routes. When an attacker intercepts packets on only one path, the original data cannot be completely obtained. The AI applications can help switches select more secure routes to forward packets, further improving the transmission

security. Another research hotspot is quantum communication, which uses quantum entanglement technology to completely eliminate the possibility of path interception. Although quantum communication remains a theory, it may have a great potential to improve transmission security.

- *Service security:* Service security does not protect packets in the network, but protects critical network nodes such as DNS servers or cloud servers. There are different servers that need to be defended, but the attack types are similar, including DDoS attacks and abnormal flow attacks. The typical defense mechanism for service security attacks is malicious packet detection and filtering, which is discussed in Section V. In fact, the interception and filtering of malicious packets also block the forwarding of normal packets, affecting the quality of service. Therefore, improving the accuracy of identification by the AI model is an issue worth studying.
- *Collaboration of Security mechanism:* Collaborative security is no longer a single security defense, but a security defense for the entire network. Collaborative security establishes a intelligent collaborative system, which can flexibly schedule various security defense mechanisms (e.g., firewall, intrusion detection system, security audit system, and log analysis system) by the AI algorithm to implement comprehensive network security defense. However, with more types of security attacks and more diversified security defense mechanisms, rapid scheduling among these mechanisms becomes increasingly difficult. Future security coordination systems with high efficiency are worth studying.

C. AI-Driven Packet Forwarding for Customized Network Scenarios

"Internet Plus" has become a new model for Internet development. The formation of intelligent information platform providing services for vertical industries is another trend of Internet development. Vertical industries are diverse and have different performance requirements for the Internet. According to characteristics of the industries, the Internet should focus on providing efficient performance in a certain aspect, such as ultra-low delay for telemedicine, network collaboration for industrial Internet, and network intelligence for smart home. The Internet and various vertical industries have formed various kinds of private networks, which plays an important role in society.

1) Challenges: The traditional Internet has relatively single functions and simple forwarding technology, which cannot provide customized network requirements for various industries. *How to meet the differentiated needs of each scenario?* This is a major challenge in the trend of network scenario customization. It is an effective solution to select appropriate Internet function modules according to the requirements of each scenario, provide corresponding service performance, and form demand-centered forwarding scheduling. In addition, it is difficult for private networks to communicate with each other. Each network scenario uses independent network protocols and packet forwarding technologies to achieve efficient internal communication. However, if a device in one scenario wants to access a device in another scenario, problems may occur. The compatibility of network protocol recognition and packet forwarding technology becomes the largest obstacle. *How to ensure information transfer across scenarios?* This poses technical challenges.

2) Research Issues: There are many kinds of network scenarios. We select some popular ones, discuss the AI-driven packet forwarding in them, and point out their research issues.

- *Telemedicine:* Telemedicine is a popular network scenario in recent years. With the COVID-19 outbreak, the need for telemedicine has become more obvious. Doctors can remotely control surgical equipment and operate on patients through a dedicated network, forming a new operation mode with no contact and zero infection. Telemedicine has a high requirement for network delay, and it is necessary to ensure that doctors at different physical locations carry out surgical operations synchronously. AI-driven packet forwarding can meet these needs. On one hand, the AI applications can select a dedicated packet transmission path to ensure the stability of surgical operations. On the other hand, the AI applications can adjust the different network delay of doctors at different physical locations to ensure the synchronization of operations. Therefore, how to further improve ultra-low delay and keep the operation synchronized are the research issues of telemedicine.
- *Internet of vehicles and intelligent transportation:* Internet of vehicles is another scenario with high requirements on network delay, which is an important part of intelligent transportation. Internet of vehicles ensures instant information exchanges between vehicles, establishes effective vehicle prediction models, and prevents traffic accidents through intelligent traffic control algorithms. Internet of vehicles and intelligent transportation also have high requirements on network computing and network reliability [162]. Vehicle network information changes frequently and rapidly. Timely information processing and strategy generation are important prerequisites to ensure intelligent transportation. The AI applications can adjust the load of data computing nodes in the vehicle network to improve computing speed. AI-driven packet forwarding can ensure high-speed and stable data transmission. However, the deployment of AI in vehicles puts high demands on the hardware, which slows down the further vehicle network development. Therefore, optimizing AI algorithms and proposing the deployment strategies are the research issues for Internet of vehicles and intelligent transportation.
- *Industrial Internet:* Industrial Internet is a platform for the effective integration of communication technology and industrial economy, which can promote the development of industry digitization, networking and intelligence. Industrial Internet focuses on network coordination ability. It connects the four systems of network, platform, data and security to provide a perfect application model for industrial production and services. Information

collaboration and integration between systems is the key to an industrial Internet. AI-driven packet forwarding technology can coordinate the work of the four systems and improve the network collaboration capability on the basis of ensuring the transmission performance. Therefore, deploying AI-driven packet forwarding for high network collaboration ability is the research issue of industrial Internet.

- *Cloud/fog computing networks:* Cloud/fog computing networks generally are an essential component of network scenarios. A cloud/fog computing network has high information computing and processing capacity, which can provide efficient data processing for many network scenarios. The deployment and allocation of resources are important issues for the development of cloud/fog computing. AI-driven packet forwarding can reduce the data transmission delay between computing nodes, balance the computing load, and improve resource allocation. Utilizing AI-driven packet forwarding for better resource deployment while maintaining a high computing performance is the research point of cloud/fog computing in the future.
- *Smart home:* Smart home focuses on the ability to generate policies and respond to network service demands. Smart home can obtain the needs of the user, generate the control instructions and realize intelligent control of the homely electrical devices. The response time of smart home program is closely related to the user experiences. The realization of fast and efficient control through AI-driven packet forwarding is the research issue of smart home.

In addition to the three research directions discussed in this section, there are other development directions for AI-driven packet forwarding, such as network resource allocation, network devices management and network fault detection. These directions are also worth studying.

VIII. Conclusion

This paper presents a survey on AI-driven packet forwarding with PDP. We first discuss the typical framework of AI-driven packet forwarding and show the problems of this framework. Then, we introduce the new framework of PDP-assisted AI-driven packet forwarding to show that PDP can improve AI-driven packet forwarding. After that, we discuss the delay, throughput, security and reliability performance improvement based on the evolution of packet forwarding: packet forwarding, AI-driven packet forwarding, and AI-driven packet forwarding with PDP, and discuss the studies on them. Finally, we elaborate our own views on the AI-driven packet forwarding evolution and propose three research directions.

The AI with PDP can effectively improve the performance of packet forwarding in many aspects, which play an important role in the development of the Internet. However, there are still many challenges in this field. This paper attempts to study the current development of packet forwarding and discusses the future research direction. We hope that our research and discussion can provide more information for other scholars to study packet forwarding and make contributions to the development of advanced networking technology.

Appendix
List of Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Network
ANTD	Abnormal Network Traffic Detection
AQM	Active Queue Management
ARIMA	Auto Regressive Integrated Moving Average
CNN	Convolutional Neural Network
DDoS	Distributed Denial of Service
DNS	Domain Name System
FIFO	First-in First-out
FPGA	Field Programmable Gate Array
ICRP	Individual Content Request Probability
IDS	Intrusion Detection System
INT	In-band Telemetry
IoT	Internet of Things
IPS	Intrusion Prevention System
KNN	K-nearest Neighbors
LFA	Link Flooding Attack
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multi-layer Perceptron
MPTCP	Multi-path TCP
PDP	Programmable Data Plane
PF	Packet Forwarding
PIFO	Push-in First-out
RF	Random Forests
SDN	Software Defined Network
SOM	Self Organizing Map
SVM	Support Vector Machine
TCA	Transfer Component Analysis
TCAM	Ternary Content Addressable Memory
URLLC	Ultra-Reliable Low-Latency Communication
VoIP	Voice over Internet Protocol
WCMP	Weighted Cost Multi-path Algorithm
XGBoost	Extreme Gradient Boosting Model

References

[1] G. Zhu and W. B. Kang, "Application and analysis of three common high-performance network data processing frameworks," in *Big Data Analytics for Cyber-Physical System in Smart City*. Heidelberg, Germany: Springer, 2021. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-33-4572-0_185

[2] J. Nam, S. Lee, H. Seo, P. Porras, V. Yegneswaran, and S. Shin, "BASTION: A security enforcement network stack for container networks," in *Proc. USENIX Annu. Technol. Conf. (USENIX ATC)*, 2020, pp. 81–95. [Online]. Available: https://www.usenix.org/conference/atc20/presentation/nam

[3] B. Rauf, H. Abbas, A. M. Sheri, W. Iqbal, and A. W. Khan, "Enterprise integration patterns in SDN: A reliable, fault-tolerant communication framework," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6359–6371, Apr. 2021.

[4] T. Jepsen, A. Fattaholmanan, M. Moshref, N. Foster, A. Carzaniga, and R. Soulé, "Forwarding and routing with packet subscriptions," in *Proc. 16th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Barcelona, Spain, Dec. 2020, pp. 282–294. [Online]. Available: https://doi.org/10.1145/3386367.3431315

[5] Z. Xiang, F. Gabriel, E. Urbano, G. T. Nguyen, M. Reisslein, and F. H. P. Fitzek, "Reducing latency in virtual machines: Enabling tactile Internet for human–machine co-working," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1098–1116, May 2019.

[6] Y. Afek, A. Bremler-Barr, and S. L. Feibish, "Zero-day signature extraction for high-volume attacks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, pp. 691–706, Apr. 2019.

[7] H. I. Abbasi, R. C. Voicu, J. A. Copeland, and Y. Chang, "Towards fast and reliable multihop routing in VANETs," *IEEE Trans. Mobile Comput.*, vol. 19, no. 10, pp. 2461–2474, Oct. 2020.

[8] J. Xie *et al.*, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, 1st Quart., 2019. [Online]. Available: https://doi.org/10.1109/COMST.2018.2866942

[9] X. Shen *et al.*, "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open J. Veh. Technol.*, vol. 1, pp. 45–66, 2020.

[10] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, 1st Quart., 2022, doi: 10.1109/COMST.2021.3135829.

[11] W. Li, H. Zhang, S. Gao, C. Xue, X. Wang, and S. Lu, "SmartCC: A reinforcement learning approach for multipath TCP congestion control in heterogeneous networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2621–2633, Nov. 2019.

[12] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer Internet of Things devices," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2018, pp. 29–35.

[13] Y. Fan, L. Wang, and X. Yuan, "Controller placements for latency minimization of both primary and backup paths in SDNs," *Comput. Commun.*, vol. 163, pp. 35–50, Nov. 2020. [Online]. Available: https://doi.org/10.1016/j.comcom.2020.09.001

[14] R. Chai, Q. Yuan, L. Zhu, and Q. Chen, "Control plane delay minimization-based capacitated controller placement algorithm for SDN," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, p. 282, Apr. 2019. [Online]. Available: https://link.springer.com/article/10.1186/s13638-019-1607-x

[15] C. Busse-Grawitz, R. Meier, A. Dietmüller, T. Bühler, and L. Vanbever. "pForest: In-Network Inference With Random Forests." 2019. [Online]. Available: http://arxiv.org/abs/1909.05680

[16] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1988–2014, 2nd Quart., 2019.

[17] P. K. Donta, T. Amgoth, and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Inf. Fusion*, vol. 49, pp. 1–25, Sep. 2019. [Online]. Available: https://doi.org/10.1016/j.inffus.2018.09.013

[18] Y. Cheng, J. Geng, Y. Wang, J. Li, D. Li, and J. Wu, "Bridging machine learning and computer network research: A survey," *CCF Trans. Netw.*, vol. 1, nos. 1–4, pp. 1–15, 2019. [Online]. Available: https://doi.org/10.1007/s42045-018-0009-7

[19] R. Bifulco and G. Rétvári, "A survey on the programmable data plane: Abstractions, architectures, and open problems," in *Proc. IEEE 19th Int. Conf. High Perform. Switching Routing (HPSR)*, 2018, pp. 1–7.

[20] E. Kaljic, A. Maric, P. Njemcevic, and M. Hadzialic, "A survey on data plane flexibility and programmability in software-defined networking," *IEEE Access*, vol. 7, pp. 47804–47840, 2019.

[21] S. Han, S. Jang, H. Choi, H. Lee, and S. Pack, "Virtualization in programmable data plane: A survey and open challenges," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 527–534, 2020.

[22] E. F. Kfoury, J. Crichigno, and E. Bou-Harb, "An exhaustive survey on P4 programmable data plane switches: Taxonomy, applications, challenges, and future trends," *IEEE Access*, vol. 9, pp. 87094–87155, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3086704

[23] L. Cui, F. R. Yu, and Q. Yan, "When big data meets software-defined networking: SDN for big data and big data for SDN," *IEEE Netw.*, vol. 30, no. 1, pp. 58–65, Feb. 2016.

[24] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Blockchain and machine learning for communications and networking systems," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1392–1431, 2nd Quart., 2020.

[25] F. R. Yu, "From information networking to intelligence networking: Motivations, scenarios, and challenges," *IEEE Netw.*, vol. 35, no. 6, pp. 209–216, Nov./Dec. 2021.

[26] H. Yao, T. Mai, X. Xu, P. Zhang, M. Li, and Y. Liu, "NetworkAI: An intelligent network architecture for self-learning control strategies in software defined networks," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4319–4327, Dec. 2018.

[27] W. Quan, M. Liu, N. Cheng, X. Zhang, D. Gao, and H. Zhang, "Cybertwin-driven DRL-based adaptive transmission scheduling for software defined vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4607–4619, May 2022.

[28] N. Foster, N. McKeown, J. Rexford, G. M. Parulkar, L. L. Peterson, and O. Sunay, "Using deep programmability to put network owners in control," *Comput. Commun. Rev.*, vol. 50, no. 4, pp. 82–88, 2020. [Online]. Available: https://doi.org/10.1145/3431832.3431842

[29] A. S. Yogapratama and M. Suryanegara, "Dealing with the latency problem to support 5G-URLLC: A strategic view in the case of an Indonesian operator," in *Proc. 2nd Int. Conf. Broadband Commun. Wireless Sensors Powering (BCWSP)*, 2020, pp. 96–100.

[30] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low overhead, fine-grained end-to-end monitoring of wireless networks using in-band telemetry," in *Proc. 15th Int. Conf. Netw. Service Manag. (CNSM)*, 2019, pp. 1–5.

[31] B. Zhang, T. Zhang, and Z. Yu, "DDoS detection and prevention based on artificial intelligence techniques," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, 2017, pp. 1276–1280.

[32] Z. Xiong and N. Zilberman, "Do switches dream of machine learning? Toward in-network classification," in *Proc. 18th ACM Workshop Hot Topics Netw. HotNets*, Princeton, NJ, USA, Nov. 2019, pp. 25–33. [Online]. Available: https://doi.org/10.1145/3365609.3365864

[33] K. A. Simpson, R. Cziva, and D. P. Pezaros, "Seior: Dataplane assisted flow classification using ML," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2020, pp. 1–6.

[34] O. Salman, I. H. Elhajj, A. Chehab, and A. Kayssi, "A machine learning based framework for IoT device identification and abnormal traffic detection," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, 2019, Art. no. e3743.

[35] K. Lee, S.-Y. Lee, and K. Yim, "Machine learning based file entropy analysis for ransomware detection in backup systems," *IEEE Access*, vol. 7, pp. 110205–110215, 2019.

[36] M. Idhammad, K. Afdel, and M. Belouch, "Semi-supervised machine learning approach for DDoS detection," *Appl. Intell.*, vol. 48, no. 10, pp. 3193–3208, 2018. [Online]. Available: https://doi.org/10.1007/s10489-018-1141-2

[37] W. He, Y. Liu, H. Yao, T. Mai, N. Zhang, and F. R. Yu, "Distributed variational Bayes-based in-network security for the Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6293–6304, Apr. 2021.

[38] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, D. Niyato, and D. I. Kim, "Distributed deep learning at the edge: A novel proactive and cooperative caching framework for mobile edge networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1220–1223, Aug. 2019.

[39] J. Vestin, A. Kassler, and J. Åkerberg, "FastReact: In-network control and caching for industrial control networks using programmable data planes," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, vol. 1, 2018, pp. 219–226.

[40] A. Filali, Z. Mlika, S. Cherkaoui, and A. Kobbane, "Preemptive SDN load balancing with machine learning for delay sensitive applications," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15947–15963, Dec. 2020.

[41] A. Alnoman, "Supporting delay-sensitive IoT applications: A machine learning approach," in *Proc. IEEE Can. Conf. Elect. Comput. Eng. (CCECE)*, 2020, pp. 1–4.

[42] L. Dan, L. Du, J. Changlin, and W. Lingqiang, "SOPA: Source routing based packet-level multi-path routing in data center networks," *ZTE Commun.*, vol. 16, no. 62, pp. 46–58, 2018.

[43] K. Liu, W. Quan, D. Gao, C. Yu, M. Liu, and Y. Zhang, "Distributed asynchronous learning for multipath data transmission based on P-DDQN," *China Commun.*, vol. 18, no. 8, pp. 62–74, 2021.

[44] M. Kheirkhah, I. Wakeman, and G. Parisis, "MMPTCP: A multipath transport protocol for data centers," in *Proc. IEEE INFOCOM 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.

[45] C. Yu *et al.*, "Reliable cybertwin-driven concurrent multipath transfer with deep reinforcement learning," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16207–16218, Nov. 2021.

[46] J. Shi *et al.*, "Flowlet-based stateful multipath forwarding in heterogeneous Internet of Things," *IEEE Access*, vol. 8, pp. 74875–74886, 2020.

[47] J. Woodruff, M. Ramanujam, and N. Zilberman, "P4DNS: In-network DNS," in *Proc. ACM/IEEE Symp. Archit. Netw. Commun. Syst. (ANCS)*, 2019, pp. 1–6.

[48] Y. Zhou *et al.*, "P4Tester: Efficient runtime rule fault detection for programmable data planes," in *Proc. IEEE/ACM 27th Int. Symp. Qual. Service (IWQoS)*, 2019, pp. 1–10.

[49] Q. Li, J. Zhang, T. Pan, T. Huang, and Y. Liu, "Data-driven routing optimization based on programmable data plane," in *Proc. 29th Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2020, pp. 1–9.

[50] A. Mubarakali, A. D. Durai, M. Alshehri, O. Alfarraj, and D. Mavaluru, "Fog-based delay-sensitive data transmission algorithm for data forwarding and storage in cloud environment for multimedia applications," *Big Data*, to be published.

[51] T. K. Patra and A. Sunny, "Forwarding in heterogeneous mobile opportunistic networks," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 626–629, Mar. 2018.

[52] M. F. Majeed, S. H. Ahmed, and M. N. Dailey, "Enabling push-based critical data forwarding in vehicular named data networks," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 873–876, Apr. 2017.

[53] K. C. Tsai, L. Wang, and Z. Han, "Mobile social media networks caching with convolutional neural network," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, 2018, pp. 83–88.

[54] P. Cheng et al., "Localized small cell caching: A machine learning approach based on rating data," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1663–1676, Feb. 2019.

[55] W.-X. Liu, J. Zhang, Z.-W. Liang, L.-X. Peng, and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075–5089, 2018.

[56] L. Luo, R. Chai, Q. Yuan, J. Li, and C. Mei, "End-to-end delay minimization-based joint rule caching and flow forwarding algorithm for SDN," *IEEE Access*, vol. 8, pp. 145227–145241, 2020.

[57] H. Huang, S. Guo, P. Li, W. Liang, and A. Y. Zomaya, "Cost minimization for rule caching in software defined networking," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 4, pp. 1007–1016, Apr. 2016.

[58] S. Bera, S. Misra, and M. S. Obaidat, "Mobi-flow: Mobility-aware adaptive flow-rule placement in software-defined access network," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1831–1842, Aug. 2019.

[59] T. Mu, A. I. Al-Fuqaha, K. Shuaib, F. Sallabi, and J. Qadir, "SDN flow entry management using reinforcement learning," *ACM Trans. Auton. Adapt. Syst.*, vol. 13, no. 2, pp. 1–11, 2018. [Online]. Available: https://doi.org/10.1145/3281032

[60] G. Grigoryan, Y. Liu, and M. Kwon, "PFCA: A programmable FIB caching architecture," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1872–1884, Aug. 2020.

[61] C. Zhang, J. Bi, Y. Zhou, K. Zhang, and Z. Ma, "B-cache: A behavior-level caching framework for the programmable data plane," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2018, pp. 84–90.

[62] S. Jung, J. Kim, and J.-H. Kim, "Intelligent active queue management for stabilized QoS guarantees in 5G mobile networks," *IEEE Syst. J.*, vol. 28, no. 4, pp. 1872–1884, Aug. 2020.

[63] C. Olariu, M. Zuber, and C. Thorpe, "Delay-based priority queueing for VoIP over software defined networks," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manag. (IM)*, 2017, pp. 652–655.

[64] T. Qiu, R. Qiao, and D. O. Wu, "EABS: An event-aware backpressure scheduling scheme for emergency Internet of Things," *IEEE Trans. Mobile Comput.*, vol. 17, no. 1, pp. 72–84, Jan. 2018.

[65] K. Zhu, G. Shen, Y. Jiang, J. Lv, Q. Li, and M. Xu, "Differentiated transmission based on traffic classification with deep learning in DataCenter," in *Proc. IFIP Netw. Conf. (Netw.)*, 2020, pp. 599–603.

[66] C. Papagianni and K. D. Schepper, "PI2 for P4: An active queue management scheme for programmable data planes," in *Proc. 15th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Orlando, FL, USA, Dec. 2019, pp. 84–86. [Online]. Available: https://doi.org/10.1145/3360468.3368189

[67] R. Kundel, J. Blendin, T. Viernickel, B. Koldehofe, and R. Steinmetz, "P4-CoDel: Active queue management in programmable data planes," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, 2018, pp. 1–4.

[68] A. G. Alcoz, A. Dietmüller, and L. Vanbever, "SP-PIFO: Approximating push-in first-out Behaviors using strict-priority queues," in *Proc. 17th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Santa Clara, CA, USA, Feb. 2020, pp. 59–76. [Online]. Available: https://www.usenix.org/conference/nsdi20/presentation/alcoz

[69] Y. Shi *et al.*, "Using machine learning to provide reliable differentiated services for IoT in SDN-like publish/subscribe middleware," *Sensors*, vol. 19, no. 6, p. 1449, 2019. [Online]. Available: https://doi.org/10.3390/s19061449

[70] C. Zhang, X. Wang, F. Li, Q. He, and M. Huang, "Deep learning-based network application classification for SDN," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 5, 2018, Art. no. e3302. [Online]. Available: https://doi.org/10.1002/ett.3302

[71] R. Parizotto *et al.*, "ShadowFS: Speeding-up data plane monitoring and telemetry using P4," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[72] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, "On load distribution over multipath networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 3, pp. 662–680, 3rd Quart., 2012.

[73] J. Zhou *et al.*, "WCMP: Weighted cost multipathing for improved fairness in data centers," in *Proc. 9th Eurosys Conf. (EuroSys)*, Apr. 2014, pp. 1–5. [Online]. Available: https://doi.org/10.1145/2592798.2592803

[74] Q. Wang, G. Shou, Y. Liu, Y. Hu, Z. Guo, and W. Chang, "Implementation of multipath network virtualization with SDN and NFV," *IEEE Access*, vol. 6, pp. 32460–32470, 2018.

[75] Y. Liu, X. Qin, T. Zhu, X. Chen, and G. Wei, "Improve MPTCP with SDN: From the perspective of resource pooling," *J. Netw. Comput. Appl.*, vol. 141, pp. 73–85, Sep. 2019. [Online]. Available: https://doi.org/10.1016/j.jnca.2019.05.015

[76] H. Zhang, J. Zhang, W. Bai, K. Chen, and M. Chowdhury, "Resilient Datacenter load balancing in the wild," in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*, Los Angeles, CA, USA, Aug. 2017, pp. 253–266. [Online]. Available: https://doi.org/10.1145/3098822.3098841

[77] L. Jin, W. Quan, G. Liu, D. Gao, C. H. Foh, and Q. Wang, "DPS: A delay-programmable scheduler for the packet out-of-order mitigation in heterogeneous networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2020, pp. 1–6.

[78] G. Liu, W. Quan, N. Cheng, N. Lu, H. Zhang, and X. Shen, "P4NIS: Improving network immunity against eavesdropping with programmable data planes," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2020, pp. 91–96.

[79] E. Vanini, R. Pan, M. Alizadeh, P. Taheri, and T. Edsall, "Let it flow: Resilient asymmetric load balancing with Flowlet switching," in *Proc. 14th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Boston, MA, USA, Mar. 2017, pp. 407–420. [Online]. Available: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/vanini

[80] K. Xue *et al.*, "DPSAF: Forward prediction based dynamic packet scheduling and adjusting with feedback for multipath TCP in lossy heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1521–1534, Feb. 2018.

[81] N. P. Katta, M. Hira, C. Kim, A. Sivaraman, and J. Rexford, "HULA: Scalable load balancing using programmable data planes," in *Proc. Symp. SDN Res. (SOSR)*, Santa Clara, CA, USA, Mar. 2016, p. 10. [Online]. Available: https://doi.org/10.1145/2890955.2890968

[82] S. T. V. Pasca, S. S. P. Kodali, and K. Kataoka, "AMPS: Application aware multipath flow routing using machine learning in SDN," in *Proc. 23rd Nat. Conf. Commun. (NCC)*, 2017, pp. 1–6.

[83] A. Azzouni, R. Boutaba, and G. Pujolle, "NeuRoute: Predictive dynamic routing for software-defined networks," in *Proc. 13th Int. Conf. Netw. Service Manag. (CNSM)*, 2017, pp. 1–6.

[84] B. Mohammed, M. Kiran, and N. Krishnaswamy, "DeepRoute on Chameleon: Experimenting with large-scale reinforcement learning and SDN on Chameleon testbed," in *Proc. IEEE 27th Int. Conf. Netw. Protocols (ICNP)*, 2019, pp. 1–2.

[85] R. Ji, Y. Cao, X. Fan, Y. Jiang, G. Lei, and Y. Ma, "Multipath TCP-based IoT communication evaluation: From the perspective of multipath management with machine learning," *Sensors*, vol. 20, no. 22, p. 6573, 2020. [Online]. Available: https://doi.org/10.3390/s20226573

[86] T. Gilad, N. R. Schiff, P. B. Godfrey, C. Raiciu, and M. Schapira, "MPCC: Online learning multipath transport," in *Proc. 16th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Barcelona, Spain, Dec. 2020, pp. 121–135. [Online]. Available: https://doi.org/10.1145/3386367.3433030

[87] M. R. Kanagarathinam, H. Natarajan, K. Arunachalam, I. Sandeep, and V. Sunil, "SMS: Smart multipath switch for improving the throughput of multipath TCP for smartphones," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2020, pp. 1–6.

[88] R. B. Basat, S. Ramanathan, Y. Li, G. Antichi, M. Yu, and M. Mitzenmacher, "PINT: Probabilistic in-band network telemetry," in *Proc. SIGCOMM Annu. Conf. ACM Special Interest Group Data Commun. Appl. Technol. Archit. Protocols Comput. Commun. Virtual Event*, Aug. 2020, pp. 662–680. [Online]. Available: https://doi.org/10.1145/3387514.3405894

[89] F. Li, J. Cao, X. Wang, and Y. Sun, "A QoS guaranteed technique for cloud applications based on software defined networking," *IEEE Access*, vol. 5, pp. 21229–21241, 2017.

[90] W.-X. Liu, "Intelligent routing based on deep reinforcement learning in software-defined data-center networks," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2019, pp. 1–6.

[91] K. Liu, "Distributed asynchronous learning for multipath data transmission based on P-DDQN," in *Proc. 3th Conf. Adv. Comput. Endogenous Security*, 2020, pp. 1–9.

[92] C. Hardegen and S. Rieger, "Prediction-based flow routing in programmable networks with P4," in *Proc. 16th Int. Conf. Netw. Service Manag. (CNSM)*, 2020, pp. 1–5.

[93] Y. Hu, Z. Li, J. Lan, J. Wu, and L. Yao, "EARS: Intelligence-driven experiential network architecture for automatic routing in software-defined networking," *China Commun.*, vol. 17, no. 2, pp. 149–162, 2020.

[94] G. Shang, P. Zhe, X. Bin, H. Aiqun, and R. Kui, "FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, 2017, pp. 1–9.

[95] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng, "XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, 2018, pp. 251–256.

[96] X. Yuan, C. Li, and X. Li, "DeepDefense: Identifying DDoS attack via deep learning," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, 2017, pp. 1–8.

[97] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based DDoS detection system in software-defined networking (SDN)," *EAI Endorsed Trans. Security Saf.*, vol. 4, no. 12, p. e2, 2017. [Online]. Available: https://doi.org/10.4108/eai.28-12-2017.153515

[98] D. Hu, P. Hong, and Y. Chen, "FADM: DDoS flooding attack detection and mitigation system in software-defined networking," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2017, pp. 1–7.

[99] T. V. Phan, N. K. Bao, and M. Park, "A novel hybrid flow-based handler with DDoS attacks in software-defined networking," in *Proc. Int. IEEE Conf. Ubiquitous Intell. Comput. Adv. Trusted Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart World Congr. (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, 2016, pp. 350–357.

[100] R. Santos, D. S. Silva, W. E. Santo, A. R. M. Ribeiro, and E. D. Moreno, "Machine learning algorithms to detect DDoS attacks in SDN," *Concurrency Comput. Pract. Exp.*, vol. 32, no. 16, 2020, Art. no. e5402. [Online]. Available: https://doi.org/10.1002/cpe.5402

[101] J. A. Pérez-Díaz, I. A. Valdovinos, K.-K. R. Choo, and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEE Access*, vol. 8, pp. 155859–155872, 2020.

[102] n. C. Lapolli, J. A. Marques, and L. P. Gaspary, "Offloading real-time DDoS attack detection to programmable data planes," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manag. (IM)*, 2019, pp. 19–27.

[103] M. Dimolianis, A. Pavlidis, and V. Maglaris, "A multi-feature DDoS detection schema on P4 network hardware," in *Proc. 23rd Conf. Innov. Clouds Internet Netw. Workshops (ICIN)*, 2020, pp. 1–6.

[104] F. Paolucci, F. Civerchia, A. Sgambelluri, A. Giorgetti, F. Cugini, and P. Castoldi, "P4 edge node enabling stateful traffic engineering and cyber security," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 11, pp. A84–A95, 2019.

[105] J. Vestin, A. Kassler, S. Laki, and G. Pongrácz, "Toward in-network event detection and filtering for publish/subscribe communication using programmable data planes," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 415–428, Mar. 2021.

[106] N. Narayanan, G. C. Sankaran, and K. M. Sivalingam, "Mitigation of security attacks in the SDN data plane using P4-enabled switches," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, 2019, pp. 1–6.

[107] F. Musumeci, V. Ionata, F. Paolucci, F. Cugini, and M. Tornatore, "Machine-learning-assisted DDoS attack detection with P4 language," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[108] Y. Mi and A. Wang, "ML-Pushback: Machine learning based pushback defense against DDoS," in *Proc. 15th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Dec. 2019, pp. 80–81. [Online]. Available: https://doi.org/10.1145/3360468.3368188

[109] X. Chen and S. Yu, "CIPA: A collaborative intrusion prevention architecture for programmable network and SDN," *Comput. Security*, vol. 58, pp. 1–19, May 2016. [Online]. Available: https://doi.org/10.1016/j.cose.2015.11.008

[110] S. Poudyal, K. P. Subedi, and D. Dasgupta, "A framework for analyzing ransomware using machine learning," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, 2018, pp. 1692–1699.

[111] H. Zhang, X. Xiao, F. Mercaldo, S. Ni, F. Martinelli, and A. K. Sangaiah, "Classification of ransomware families with machine learning based on N-gram of opcodes," *Future Gener. Comput. Syst.*, vol. 90, pp. 211–221, Jan. 2019. [Online]. Available: https://doi.org/10.1016/j.future.2018.07.052

[112] O. M. K. Alhawi, J. Baldwin, and A. Dehghantanha. "Leveraging Machine Learning Techniques for Windows Ransomware Network Traffic Detection." 2018. [Online]. Available: http://arxiv.org/abs/1807.10440

[113] S. K. Shaukat and V. J. Ribeiro, "RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning," in *Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, 2018, pp. 356–363.

[114] G. Cusack, O. Michel, and E. Keller, "Machine learning-based detection of ransomware using SDN," in *Proc. ACM Int. Workshop Security Softw. Defined Netw. Netw. Function Virtualization (SDN-NFVSec@CODASPY)*, Tempe, AZ, USA, Mar. 2018, pp. 1–6. [Online]. Available: https://doi.org/10.1145/3180465.3180467

[115] B. Kong, Z. Liu, G. Zhou, and X. Yu, "A method of detecting the abnormal encrypted traffic based on machine learning and Behavior characteristics," in *Proc. 9th Int. Conf. Commun. Netw. Security (ICCNS)*, Chongqing, China, Nov. 2019, pp. 47–50. [Online]. Available: https://doi.org/10.1145/3371676.3371705

[116] Y. Zhang, X. Chen, D. Guo, M. Song, Y. Teng, and X. Wang, "PCCN: Parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows," *IEEE Access*, vol. 7, pp. 119904–119916, 2019.

[117] L. Boero, M. Marchese, and S. Zappatore, "Support vector machine meets software defined networking in IDS domain," in *Proc. 29th Int. Teletraffic Congr. (ITC)*, vol. 3, 2017, pp. 25–30.

[118] S. S. Bhunia and M. Gurusamy, "Dynamic attack detection and mitigation in IoT using SDN," in *Proc. 27th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, 2017, pp. 1–6.

[119] L. Kong, G. Huang, and K. Wu, "Identification of abnormal network traffic using support vector machine," in *Proc. 18th Int. Conf. Parallel Distrib. Comput. Appl. Technol. (PDCAT)*, 2017, pp. 288–292.

[120] C. Song, Y. Park, K. Golani, Y. Kim, K. Bhatt, and K. Goswami, "Machine-learning based threat-aware system in software defined networks," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2017, pp. 1–9.

[121] G. A. Ajaeiya, N. Adalian, I. H. Elhajj, A. Kayssi, and A. Chehab, "Flow-based intrusion detection system for SDN," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2017, pp. 787–793.

[122] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software defined networking," in *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, 2016, pp. 258–263.

[123] S. Ji, B. Jeong, S. Choi, and D. H. Jeong, "A multi-level intrusion detection method for abnormal network behaviors," *J. Netw. Comput. Appl.*, vol. 62, pp. 9–17, Feb. 2016. [Online]. Available: https://doi.org/10.1016/j.jnca.2015.12.004

[124] J. Niu, Y. Zhang, D. Liu, D. Guo, and Y. Teng, "Abnormal network traffic detection based on transfer component analysis," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2019, pp. 1–6.

[125] H. Kim and A. Gupta, "ONTAS: Flexible and scalable online network traffic anonymization system," in *Proc. Workshop Netw. Meets AI ML NetAI@SIGCOMM*, Beijing, China, Aug. 2019, pp. 15–21. [Online]. Available: https://doi.org/10.1145/3341216.3342208

[126] J. Hypolite, J. Sonchack, S. Hershkop, N. Dautenhahn, A. DeHon, and J. M. Smith, "DeepMatch: Practical deep packet inspection in the data plane using network processors," in *Proc. 16th Int. Conf. Emerg. Netw. EXperiments Technol. (CoNEXT)*, Barcelona, Spain, Dec. 2020, pp. 336–350. [Online]. Available: https://doi.org/10.1145/3386367.3431290

[127] D. Scholz, S. Gallenmüller, H. Stubbe, B. Jaber, M. Rouhi, and G. Carle. "Me Love (SYN-)Cookies: SYN Flood Mitigation in Programmable Data Planes." 2020. [Online]. Available: https://arxiv.org/abs/2003.03221

[128] R. Negi, A. Dutta, A. Handa, U. Ayyangar, and S. K. Shukla, "Intrusion detection amp; Prevention in programmable logic controllers: A model-driven approach," in *Proc. IEEE Conf. Ind. Cyberphysical Syst. (ICPS)*, vol. 1, 2020, pp. 215–222.

[129] L. Castanheira, R. Parizotto, and A. E. Schaeffer-Filho, "FlowStalker: Comprehensive traffic flow monitoring on the data plane using P4," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.
[130] J. H. Lee and K. Singh, "SwitchTree: In-network computing and traffic analyses with random forests," in *Neural Computing and Applications*. Heidelberg, Germany: Springer, 2020.
[131] R. U. Rasool, H. Wang, U. Ashraf, K. Ahmed, Z. Anwar, and W. Rafique, "A survey of link flooding attacks in software defined network ecosystems," *J. Netw. Comput. Appl.*, vol. 172, Dec. 2020, Art. no. 102803. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1084804520302757
[132] R. U. Rasool, U. Ashraf, K. Ahmed, H. Wang, W. Rafique, and Z. Anwar, "Cyberpulse: A machine learning based link flooding attack mitigation system for software defined networks," *IEEE Access*, vol. 7, pp. 34885–34899, 2019.
[133] T. M. Hoang, N. M. Nguyen, and T. Q. Duong, "Detection of eavesdropping attack in UAV-aided wireless systems: Unsupervised learning with one-class SVM and K-means clustering," *IEEE Wireless Commun. Lett.*, vol. 9, no. 2, pp. 139–142, Feb. 2020.
[134] L. Xiao, G. Sheng, S. Liu, H. Dai, M. Peng, and J. Song, "Deep reinforcement learning-enabled secure visible light communication against eavesdropping," *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 6994–7005, Oct. 2019.
[135] M. Liu *et al.*, "Learning based adaptive network immune mechanism to defense eavesdropping attacks," *IEEE Access*, vol. 7, pp. 182814–182826, 2019.
[136] P. Janakaraj, P. Pinyoanuntapong, P. Wang, and M. Lee, "Towards in-band telemetry for self driving wireless networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2020, pp. 766–773.
[137] J. Vestin, A. Kassler, D. Bhamare, K.-J. Grinnemo, J.-O. Andersson, and G. Pongracz, "Programmable event detection for in-band network telemetry," in *Proc. IEEE 8th Int. Conf. Cloud Netw. (CloudNet)*, 2019, pp. 1–6.
[138] Y. Zhou *et al.*, "Newton: Intent-driven network traffic monitoring," in *Proc. 16th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, Barcelona, Spain, Dec. 2020, pp. 295–308. [Online]. Available: https://doi.org/10.1145/3386367.3431298
[139] T. Holterbach, E. C. Molero, M. Apostolaki, A. Dainotti, S. Vissicchio, and L. Vanbever, "BLINK: Fast connectivity recovery entirely in the data plane," in *Proc. 16th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Boston, MA, USA, Feb. 2019, pp. 161–176. [Online]. Available: https://www.usenix.org/conference/nsdi19/presentation/holterbach
[140] A. Lazaris and V. K. Prasanna, "DeepFlow: A deep learning framework for software-defined measurement," in *Proc. 2nd Workshop Cloud-Assisted Netw. (CAN@CoNEXT)*, Incheon, Republic of Korea, Dec. 2017, pp. 43–48. [Online]. Available: https://doi.org/10.1145/3155921.3155922
[141] R. Hohemberger *et al.*, "Orchestrating in-band data plane telemetry with machine learning," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2247–2251, Dec. 2019.
[142] F. Yang, W. Quan, N. Cheng, Z. Xu, X. Zhang, and D. Gao, "Fast-INT: Light-weight and efficient in-band network telemetry in programmable data plane," in *Proc. IEEE 92nd Veh. Technol. Conf. (VTC-Fall)*, 2020, pp. 1–5.
[143] K. S. Mayer, J. A. Soares, R. P. Pinto, C. E. Rothenberg, D. S. Arantes, and D. A. A. Mello, "Machine-learning-based soft-failure localization with partial software-defined networking telemetry," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 13, pp. E122–E131, 2021.
[144] A. Pashamokhtari, "Ph.D. forum abstract: Dynamic inference on IoT network traffic using programmable telemetry and machine learning," in *Proc. 19th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, 2020, pp. 371–372.
[145] J. Hyun, N. Van Tu, and J. W.-K. Hong, "Towards knowledge-defined networking using in-band network telemetry," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp. (NOMS)*, 2018, pp. 1–7.
[146] S. Jain, M. Khandelwal, A. Katkar, and J. Nygate, "Applying big data technologies to manage QoS in an SDN," in *Proc. 12th Int. Conf. Netw. Service Manag. (CNSM)*, 2016, pp. 302–306.
[147] Z. Zhou, S. Chahal, T.-Y. Ho, and A. Ivanov, "Supervised-learning congestion predictor for routability-driven global routing," in *Proc. Int. Symp. VLSI Design Autom. Test (VLSI-DAT)*, 2019, pp. 1–4.
[148] A. Feldmann, B. Chandrasekaran, S. Fathalli, and E. N. Weyulu, "P4-enabled network-assisted congestion feedback: A case for NACKs," in *Proc. Workshop Buffer Sizing (BS)*, Dec. 2019, pp. 1–3. [Online]. Available: https://doi.org/10.1145/3375235.3375238
[149] T. Mai, H. Yao, X. Zhang, Z. Xiong, and D. Niyato, "A distributed reinforcement learning approach to in-network congestion control," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, 2020, pp. 817–822.
[150] J. Li, Y. Guan, P. Ding, and S. Wang, "TCP-PPCC: Online-learning proximal policy for congestion control," in *Proc. 21st Asia–Pac. Netw. Oper. Manag. Symp. (APNOMS)*, 2020, pp. 243–246.
[151] J. Liu *et al.*, "P4V: Practical verification for programmable data planes," in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*, Budapest, Hungary, Aug. 2018, pp. 490–503. [Online]. Available: https://doi.org/10.1145/3230543.3230582
[152] R. Stoenescu, D. Dumitrescu, M. Popovici, L. Negreanu, and C. Raiciu, "Debugging P4 programs with vera," in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*, Aug. 2018, pp. 518–532. [Online]. Available: https://doi.org/10.1145/3230543.3230548
[153] L. Freire, M. C. Neves, L. Leal, K. Levchenko, A. E. S. Filho, and M. P. Barcellos, "Uncovering bugs in P4 programs with assertion-based verification," in *Proc. Symp. SDN Res. (SOSR)*, Los Angeles, CA, USA, Mar. 2018, pp. 1–4. [Online]. Available: https://doi.org/10.1145/3185467.3185499
[154] P. Zhang, "Towards rule enforcement verification for software defined networks," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.
[155] C. Zhang *et al.*, "P4DB: On-the-fly debugging of the programmable data plane," in *Proc. IEEE 25th Int. Conf. Netw. Protocols (ICNP)*, 2017, pp. 1–10.
[156] L. J. Jagadeesan and V. Mendiratta, "Analytics-enhanced automated code verification for dependability of software-defined networks," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops*, 2017, pp. 1–9.
[157] A. Shukla, K. N. Hudemann, A. Hecker, and S. Schmid, "Runtime verification of P4 switches with reinforcement learning," in *Proc. Workshop Netw. Meets AI ML NetAI@SIGCOMM*, Beijing, China, Aug. 2019, pp. 1–7. [Online]. Available: https://doi.org/10.1145/3341216.3342206
[158] H. Zhang, W. Quan, H.-C. Chao, and C. Qiao, "Smart identifier network: A collaborative architecture for the future Internet," *IEEE Netw.*, vol. 30, no. 3, pp. 46–51, May/Jun. 2016.
[159] H. Zhang and W. Quan, "Networking automation and intelligence: A new era of network innovation," *Engineering*, to be published.
[160] J. Kang *et al.*, "Communication-efficient and cross-chain empowered federated learning for artificial intelligence of things," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 2966–2977, Sep./Oct. 2022.
[161] D. C. Morales *et al.*, "Blockgraph proof-of-concept," in *Proc. SIGCOMM ACM SIGCOMM Conf. Virtual Event*, Aug. 2021, pp. 82–84. [Online]. Available: https://doi.org/10.1145/3472716.3472866
[162] M. Xu, D. T. Hoang, J. Kang, D. Niyato, Q. Yan, and D. I. Kim, "Secure and reliable transfer learning framework for 6G-enabled Internet of Vehicles," *IEEE Wireless Commun.*, vol. 29, no. 4, pp. 132–139, Aug. 2022.

Wei Quan (Senior Member, IEEE) received the Ph.D. degree in communication and information system from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014.

He is currently a Full Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing. He has authored or coauthored more than 50 papers in prestigious international journals and conferences, including *IEEE Communications Magazine*, IEEE WIRELESS COMMUNICATIONS, IEEE NETWORK, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE COMMUNICATIONS LETTERS, *IFIP Networking*, IEEE ICC, and IEEE GLOBECOM. His research interests include key technologies for network analytics, future Internet, 6G networks, and vehicular networks.

Dr. Quan is the TPC Member of IEEE Globecom (2019–2022), IEEE ICC (2017–2020), IEEE INFOCOM (NewIP Workshop) 2020, and ACM MOBIMEDIA (2015–2017). He is an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Peer-to-Peer Networking and Applications*, *Journal of Internet Technology*, and IEEE ACCESS and as a technical reviewer for many important international journals. He is also a member of ACM and a Senior Member of the Chinese China Institute of Communications.

Ziheng Xu received the B.E. degree in communication engineering in 2019. He is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China.

His research interests include software-defined networking, concurrent multipath transfer, machine learning, high-speed railway networks, and programmable data planes.

Mingyuan Liu received the bachelor's degree in communication engineering, in 2018. He is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China.

His research interests include future networks, software defined networking, Internet of Things, and cybersecurity.

Nan Cheng (Member, IEEE) received the B.E. and M.S. degrees from the Department of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2009 and 2012, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2016.

From 2017 to 2019, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. He is currently a Professor with the State Key Laboratory of ISN and the School of Telecommunications Engineering, Xidian University, Xi'an, China. He has authored or coauthored more than 70 journal papers in IEEE Transactions and other top journals. His current research interests include B5G/6G, space-air-ground integrated network, big data in vehicular networks, and self-driving system. His research focuses on applying AI techniques for future networks.

Prof. Cheng is/was the guest editor of several journals and an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, and *Peer-to-Peer Networking and Applications*.

Gang Liu (Student Member, IEEE) received the B.E. degree from Tiangong University, Tianjin, China, in 2016, and the Ph.D. degree with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China, in 2021.

He is currently an Engineer with China Telecom Research Institute, Shanghai, China. His current research interests include information-centric networking, software-defined networking, network function virtualization, programmable network language, and stateful forwarding.

Deyun Gao (Senior Member, IEEE) received the B.E. and M.E. degrees in electrical engineering and the Ph.D. degree in computer science from Tianjin University, Tianjin, China, in 1994, 1999, and 2002, respectively.

He spent one year as a Research Associate with the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Hong Kong. Then, he spent three years as a Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. In 2007, he joined the Faculty of School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing, China, as an Associate Professor and was promoted to a Full Professor, in 2012. In 2014, he was a Visiting Scholar with the University of California at Berkeley, Berkeley, CA, USA. His research interests include Internet of Things, vehicular networks, and next-generation Internet.

Hongke Zhang (Fellow, IEEE) received the Ph.D. degree in communication and information system from the University of Electronic Science and Technology of China, Chengdu, China, in 1992.

He is currently a Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China, where he currently directs the National Engineering Center of China on Mobile Specialized Network. He is an Academician of China Engineering Academy and the Co-Director of the PCL Research Center of Networks and Communications, Peng Cheng Laboratory. His current research interests include architecture and protocol design for the future Internet and specialized networks.

Prof. Zhang currently serves as an Associate Editor for the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT and IEEE INTERNET OF THINGS JOURNAL.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is an University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks. He served as the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *IET Communications*. He was the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globeco'07, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is a Fellow of the Engineering Institute of Canada, Canadian Academy of Engineering, and Royal Society of Canada, and a Foreign Member of Chinese Academy of Engineering. He is the President of IEEE Communications Society.

Weihua Zhuang (Fellow, IEEE) is an University Professor and the Tier I Canada Research Chair of Wireless Communication Networks with the University of Waterloo, Canada. Her research focuses on network architecture, algorithms and protocols, and service provisioning in future communication systems. She was the Editor-in-Chief of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2007 to 2013, the General Co-Chair of 2021 IEEE/CIC International Conference on Communications in China, the Technical Program Chair/Co-Chair of IEEE VTC 2017 Fall/2016 Fall, the Technical Program Symposia Chair of 2011 IEEE Globecom, and an IEEE Communications Society Distinguished Lecturer from 2008 to 2011. She is an Elected Member of the Board of Governors and the Executive Vice President of the IEEE Vehicular Technology Society. She is a Fellow of Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada.