This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal

An Intelligent Traffic Load Prediction Based Adaptive Channel Assignment Algorithm in SDN-IoT: A Deep Learning Approach

Fengxiao Tang, *Student Member, IEEE,* Zubair Md. Fadlullah, *Senior Member, IEEE,* Bomin Mao, *Student Member, IEEE,* and Nei Kato, *Fellow, IEEE*.

***Abstract*—Due to the fast increase of sensing data and quick response requirement in the Internet of Things (IoT) delivery network, the high speed transmission has emerged as an important issue. Assigning suitable channels in the wireless IoT delivery network is a basic guarantee of high speed transmission. However, the high dynamics of traffic load make the conventional fixed channel assignment algorithm ineffective. Recently, the Software Defined Networking based IoT (SDN-IoT) is proposed to improve the transmission quality. Besides this, the intelligent technique of deep learning is widely researched in high computational SDN. Hence, we firstly propose a novel deep learning based traffic load prediction algorithm to forecast future traffic load and congestion in network. Then, a Deep Learning based Partially Channel Assignment Algorithm, referred to as DLPOCA, is proposed to intelligently allocate channels to each link in the SDN-IoT network. Finally, we consider a deep learning based prediction and Partially Overlapping Channel Assignment (POCA) to propose a novel intelligent channel assignment algorithm (TP-DLPOCA), which can intelligently avoid potential congestion and quickly assign suitable channels in SDN-IoT. The simulation result demonstrates that our proposal significantly outperforms conventional channel assignment algorithms.**

***Index Terms*—Deep learning, Internet of Things (IoT), Software Defined Network (SDN), Partially Overlapping Channel Assignment (POCA), traffic load prediction.**

I. Introduction

Real world Internet of Things (IoT) [1]–[4] deployments are fundamentally heterogeneous. Software Defined Networking (SDN) [5], [6] is a famous technique used in the IoT to deal with heterogeneous resources and structures [7]. In such SDN-IoT as depicted in Fig. 1, heterogeneous devices sense and collect data in the sensing plane, and then send the data to the gateway after integration through switches in the data plane. With the increasing number of devices, the load of integrated traffic in switches may become significantly heavy, and multiple channels are needed to be evenly assigned to each link to balance the load [8]–[10]. Since high interference exists between non-orthogonal channels and the number of orthogonal channels is limited, the Partially Overlapping Channel (POC) can be a good solution to decrease interference and improve network throughput [11]–[13]. However, current POC algorithms mostly focus on the improvement of network performance after channel assignment, but lack the consideration of waste throughput due to the suspended transmission during the channel assignment process. With the high dynamics of the current IoT, the assigned channels need to be frequently changed to adaptively adjust to the dynamically changed network traffic. This dynamic adjustment throws out a critical requirement for the quick processing of the channel assignment. To solve this problem, in our previous work [14], an Anti-Coordination based POC Algorithm (ACPOCA) was proposed, which can efficiently reduce the iteration times of channel assignment process, and improve the network throughput. However, without a central controller, both the signaling and suspension time of the network are limited by the distributed setting. Therefore, to address such challenges, in the first part of this paper, a deep learning based, intelligent POC assignment algorithm with the centralized SDN is proposed. The **contributions** of the deep learning based proposal, in the first part of the paper, can be explained in two aspects.

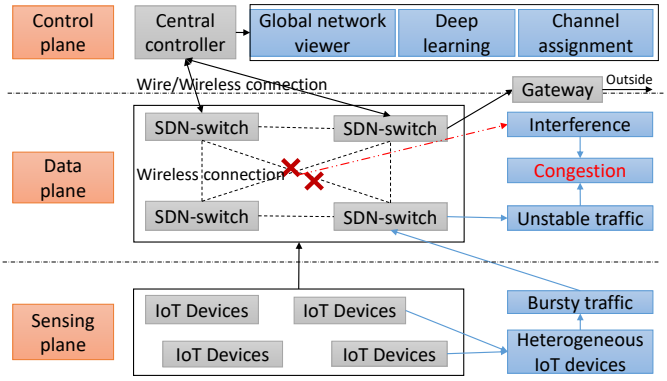


Fig. 1: The SDN-IoT architecture.

- First, with the central control paradigm of SDN, switches do not need to exchange their channel states anymore. All channel assignment processes can be carried out in the central controller. Thus, the signaling overhead of the network is significantly reduced.
- Second, since the deep learning approach can learn from previous channel assignment processes through training with the data collected from the existing channel assignment algorithms (e.g., ACPOCA), the channel assignment can be finished in just single iteration.

In summary, this approach, which we refer to as the Deep Learning based Channel Assignment (DLCA), can efficiently

F. Tang, Z. M. Fadlullah, B. Mao, and N. Kato are with the Graduate School of Information Sciences (GSIS), Tohoku University, Sendai, Japan. Emails: {fengxiao.tang, zubair, bomin.mao, kato}@it.is.tohoku.ac.jp

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal

reduce the suspension time caused by channel assignment, and achieves almost non-suspending flows during the channel assignment process.

Additionally, in existing channel assignment algorithms, as the most important baseline metric in the channel assignment, the traffic load is usually assumed to be continuous and stable. This means that the traffic load in the next time interval after the channel assignment is similar to that in last time interval. However, the real traffic loads in practical networks are more complex and may suddenly change like a bursty traffic. Particularly in SDN-IoT, in the sensing plane of the SDN-IoT structure, the devices can be divided into three groups depending on the sensing mechanism: periodic sensing, event-driven sensing, and query-based sensing [15]–[18]. For the periodic sensing devices, such as temperature, humidity and light sensing devices, they sense data and periodically integrate and transmit them to the central controller. Moreover, these devices may have different policies (e.g., sensing circle, volume of sensing data, and so forth). For example, a kind of temperature sensing device may collect 3kB temperature once in every 30s, another kind of humidity sensing device may collect 10kB humidity data once every 1 minute. Those different policies result in highly complex, periodically bursty distribution of the traffic load. For the event-driven and query-based sensing devices, the traffic load is also not consequent but explosively generated when a new event occurs or a query comes. The bursty traffic caused by the event-driven and query-based sensing device is more random and irregular than that generated by the periodic sensors. In the SDN-IoT network, with heterogeneous resources, sensing devices can hardly cooperate with one another, making the switch impossible to know the real future traffic integrated by the heterogeneous sensors. Furthermore, besides the traffic generated by its connected sensing devices represented as integrated traffic, each switch may also have to forward the traffic from other switches denoted as relayed traffic. In practical networks, the mixed traffic containing integrated and relayed traffic becomes more complex. Even though many existing researchers proposed some methods about traffic load prediction, most of them focused on the traffic changes in the long-term and did not consider the traffic load changes caused by routing. Therefore, in the second part of this this paper, a deep learning based prediction and POCs assignment algorithm is proposed. The **contributions** of the second part of the paper are separately outlined next.

- First, we use the powerful deep learning approach to predict the complex traffic, which can achieve above 90 percent accuracy and have a quick response time. (5ms<)
- Second, we investigate the advantage of using the centralized SDN technique in the deep learning based traffic load prediction in the IoT environment. In order to show the improvement of deep learning based traffic load prediction in SDN-IoT compare with conventional IoT, we respectively design three traffic load prediction algorithms to suit three different control systems (i.e., centralized SDN control system, simi-centralized control system and distributed control system). After designing those three different prediction methods, we further compare the prediction accuracy in those three different control systems. The result shows that, the prediction accuracy of centralized SDN based prediction is always better than those in the two other systems.
- Finally, with the centralized SDN control, we combine the deep learning based traffic prediction and channel assignment, that uses the predicted traffic load as the criterion to perform the intelligent channel assignment. Such proposed intelligent channel assignment, which we refer to as TP-DLCA, can efficiently increase the channel assignment accuracy and processing speed of channel assignment. The simulation results demonstrate that both the throughput and delay in the SDN-IoT with our proposal are better than those of the conventional algorithms.

The remainder of this paper is organized as follows. Sec. II surveys related works. Sec. III presents the considered network architecture, interference model, and training model. Then, In Sec. IV, we propose a deep learning based traffic load prediction method by considering different network control systems. Then, we propose the deep learning based channel assignment algorithm, and combine it with deep learning based traffic load prediction in Sec. V-A. The performance evaluation of our proposals are presented in Sec. VI. Finally, Sec. VII concludes the paper.

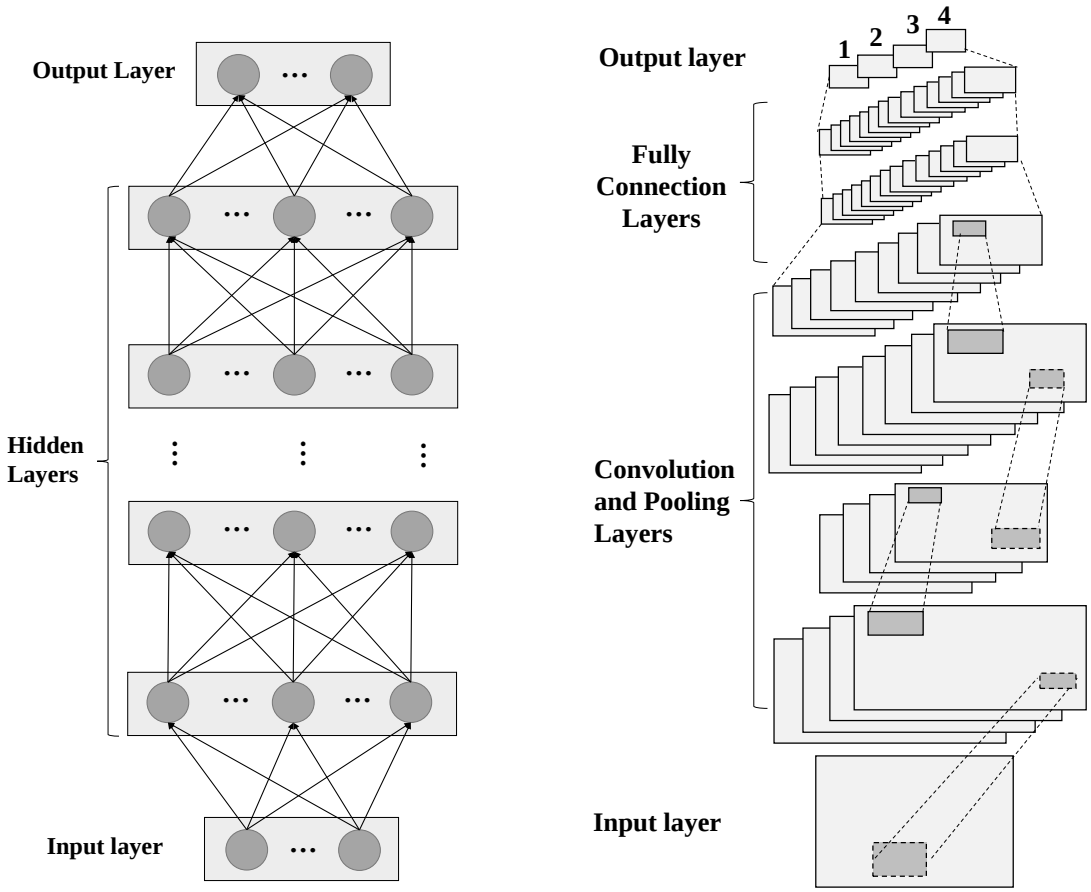
II. Related research works

The SDN-IoT structure was first proposed by Qin at al. in 2014 [7] which incorporates and supports commands in a heterogeneous structure to optimize the SDN-IoT network. After the first SDN-IoT structure was proposed, many research works emerged. Sood et al [19] employed a multi-objective constraint to manage the layer resource in SDN-IoT. Ojo et al [20] presented a SDN-IoT architecture with Network Function Virtualization (NFV) implementation to address the new challenges of IoT. Nguyen et al [21] proposed a SDN-based IoT Mobile Edge Cloud Architecture to deploy diverse IoT services at the mobile edge.

The work in [22] proposed a new channel assignment strategy based on non-overlapping channels, and demonstrated how this contributes to spectrum utilization and improves the bandwidth available to the network users. On the other hand, the works in [12], [13] demonstrated that the use of overlapping channels leads to better performance in contrast to three non-overlapping channels for wireless networks. Following this finding, new heuristic channel assignment algorithms were proposed in [23]. In our earlier work in [11], instead of heuristics, an optimal channel assignment exploiting POCs for wireless mesh networks was proposed. After that, we further proposed an Anti-Coordination Game based Partially Overlapping Channels Assignment, referred to as ACPOCA, [14], in which, the nodes use only local information to play the game, and can reach a steady state quickly. Moreover, since this ACPOCA is dynamic, it is suitable to be employed in the fixed or dynamic topology network.

Our earlier work in [24] envisioned the first proof-of-concept of using deep learning architectures for substantially

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal



(a) Considered DBA structure.

(b) Considered deep CNN structure.

Fig. 2: The employed deep learning structures.

improving the heterogeneous network traffic control. A deep learning system was proposed that can be trained in a supervised manner based on uniquely characterized inputs using traffic patterns at the edge routers of a wireless backbone network. However, the deep learning algorithm was trained upon a considered benchmark routing method, namely Open Shortest Path First (OSPF). The survey conducted in [25] demonstrated that there exist different deep learning architectures such as Deep Boltzmann Machines (DBMs), Deep CNNs, and so forth that could be exploited for network traffic control systems. However, the case study considered in that work also considered a baseline routing method for training the deep learning algorithm. Furthermore, the work in [26] explored current Software Defined Router (SDR) architectures and demonstrated how the deep learning technique can be harnessed to compute the routing paths. The Graphics Processing Unit (GPU)-accelerated SDR enabling massively parallel computing for the deep learning was shown to substantially improve the backbone network traffic control. However, similar to the afore-mentioned researches, this work also adopted a supervised deep learning system dependent on a conventional rule-based routing method. Then, our previous work in [27] proposed a real-time deep learning approach for intelligent traffic control in the wireless network exploiting deep CNNs with uniquely characterized inputs and outputs to intelligently control the network packets forwarding. However, all of the previous researches focused on the intelligent packets forwarding process and did not take into account the intelligent channel assignment problem in the wireless network.

III. System model

In this section, we describe our considered network model, interference and training process.

A. Network Model

Consider the SDN-IoT is constructed in a heterogeneous structure which contains different kinds of devices. Devices sense and collect data, and then send the data to the gateway through multiple switches. For better understanding, we use graph $\mathbf{G} = (D \cup S \cup C, E)$ to represent the network where D denotes the set of devices in the network and $D = \{d_1, d_2, \ldots, d_{|D|}\}$. And S denotes the set of switches and $S = \{s_1, s_2, \ldots, s_M\}$ where M is the total number of switches. Consider the switches are randomly deployed in the considered area, and each switch serves the devices located in its own service area. For example, an Access Point (AP) of a residence is regarded as a switch and all the devices in this house are served by the AP. The average number of devices belong to each switch area is presented as R, namely

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal

$|D| = M \times R$. Each switch collects data from devices, and then send them to the gateway with multi-hop transmission. The central controller C is deployed randomly in the network as the global network viewer to manage all packets forwarding, deep learning process, channel assignment and other network problems. The structure of the SDN-IoT is shown as Fig.1.

In the sensing plane, we consider Q different kinds of periodic sensing devices and W different kinds of event driven sensing devices with totally number of $|D|$ deployed in the whole area. For example, one kind of periodic sensing device senses and collects 10kB data in every 30s and another kind of periodic sensing device collects 7kB data in every 20s.

Let E represents the edges set in the graph $\mathbf{G}$. Furthermore, the edge $e \in E$ in the graph means the link between two vertices. The weight, $w(e)$, represents the connection ability of the link e. This weight depends on many factors such as the transmission distance, transmission power, interference, bandwidth, and so on. Consider the links between devices and switches use different spectrum from the links between switches. The data sensed by a single device are small and the capacity requirement of a single link between devices and switches is not so strict. Therefore, the considered interference mainly exists in the links between the switches in the data plane.

B. Interference Model

In a multi-channel deployment network, the interference between each channel is critical to the network capacity. In this section, we present an interference range vector based model to state those interferences.

Consider the channel interference between channels is affected by both geographical distance and distance in the spectrum [11]. In order to calculate the interference affected by geographical distance, an interference range vector, shown in Table I, is used to measure the interference level of channels [28]. Here δ refers to the interference range for a channel separation between channels p and q, $IR(\delta) = |p-q|$ denotes the geographical interference distance between channels p and q.

TABLE I: Interference Range (IR).

δ	0	1	2	3	4	5
$IR(\delta)$	132.6	90.8	75.9	46.9	32.1	0

Then, consider the different geographical distance between assigned channels. We use a weight metric *Interference Factor* $f_{p,q}$ to denote the interference level between two active channels p and q assigned to links.

Now, let d refer to the distance between nodes operating with channels p and q. If the nodes use the same channel, d is set to zero. Then, $f_{p,q}$ is calculated in the following three cases respectively:

1) $f_{p,q} = 0$: when $\delta \geq 5$ or $d > IR(\delta)$.
   When the nodes are assigned orthogonal channels or have enough distance to avoid interference, no interference occurs between the radios.
2) $1 < f_{p,q} < \infty$: when $0 \leq \delta < 5$ and $d \leq IR(\delta)$.
   When overlapping interference occurs, the distance between the nodes is smaller than the interference range. In this case, IF should be a ratio proportional to the distance between the nodes. IF can be calculated as follows:
   $$f_{p,q} = IR(\delta)/d. \tag{1}$$
3) $f_{p,q} = \infty$: when $0 \leq \delta < 5$ and $d = 0$.
   This happens because of the self interference problem. Hence, two overlapping channels ($\delta < 5$) are not viable to be assigned to the node due to their full interference.

Here, $f_{p,q}$ is used to measure the interference level of channels, and is the main parameter used in the POC assignment algorithm to judge the quality of POCs. In order to quickly measure all the channels conditions, in the conventional partially channel assignment algorithms, each router uses the interference matrix ($IMatrix$) to record the $f_{p,q}$ value of all the links. And all routers need to broadcast their channel information and update $IMatrix$ continuously, which result in large signaling. On the other hand, in our proposed deep learning based channel assignment algorithm, the $IMatrix$ is no longer needed, each switch just receives traffic load information and activates neural network weight matrix obtained by the training process. The only signaling overhead is the traffic load transmission process between switch and central controller. Next, we describe the deep learning training model used in training process.

C. Training Model

In our proposed training process, we consider two neural network structures, the basic Deep Belief Architecture (DBA) and the deep Convolutional Neural Network (CNN). As shown in Fig. 2(a), the chosen DBA is constructed with L layers, including one input layer, one visible output layer and $(L-2)$ hidden layers. The unit in each layer except the input layer has its own weight value called bias. And the units in two adjacent layers are connected with each other via weighted links while no inner layer connection exists. Let x_{input} and y_{output} denote the values of units in the input and output layer, respectively. w_{ij} denotes the weight of link between units i and j, and b_i represents the bias of unit i. Additionally, w and b represent the matrices consisting weights of all links and all the bias values, respectively. The training of the DBA consists of two steps, namely forward propagation and back propagation processes. The forward propagation is used to construct the structure and activate output, while the back propagation is used to adapt the structure and fine-tune the values of w and b. As modeled in our previous work in [26], the forward propagation process can be modeled as a log-likelihood function,

$$l(w, b, x_{input}, y_{output}) = \sum_{t=1}^{m} \log p(\boldsymbol{v}^{(t)}), \tag{2}$$

where, $\boldsymbol{v}^{(t)}$ denotes the t^{th} training data. The DBA training process can be seen as a log-linear Markov Random Field (MRF). Hence, we use $p(\boldsymbol{v}^{(t)})$ to represent the probability of $\boldsymbol{v}^{(t)}$. Here, m represents the total number of training data.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal

Since the purpose of the training process is to maximize $l(w, b, x_{input}, y_{output})$, in the backpropagation process, the gradient descent method is adopted to adjust the link weight w and bias b, which is represented as:

$$w = w + \eta \frac{\partial l(w, b, x_{input}, y_{output})}{\partial w}, \tag{3}$$

$$b = b + \eta \frac{\partial l(w, b, x_{input}, y_{output})}{\partial b}, \tag{4}$$

where, η is the learning rate of training process.

The second considered deep learning structure is the deep CNN as shown in Fig. 2(b). At the first glance, the structure of deep CNN is similar to DBA, and the main training process also includes forward and back propagation. However, when the size of the input layer becomes quite large and spatially connected in high dimensions, the DBA cannot capture the spatial features efficiently. As a powerful deep learning structure, the deep CNN is widely used in image identification and natural language processing [29], [30]. In the deep CNN, the covolutional layers are good at capture the spatial and temporal connections of the input data. [31]. This is a better choice to construct the learning system of centralized network of spatial connection extraction. To better extract the spatial connections of input data, the convolutional and pooling layers are employed in the deep CNN. The convolution operation is used to filter the input and pass the result to the next layer, while the pooling layers are used to combine the outputs of the neuron clusters at one layer into a single neuron in the next layer, which can further reduce the redundant data and extract the wide range spatial features. Different filters may be used in each convolutional layer and their results are combined to transfer to fully connection layers. With the utilization of convolutional and pooling layers, the features of input can be efficiently extracted, which significantly reduces the computation burden.

As the purpose of the convolution operation is to extract the distinguished features of the input, the parameters (weights and biases) of the convolution operation consist of a set of learnable filters. If we use $W^{(l_1)}$ to denote the filters and the k^{th} filter is represented by $W_k^{(l_1)}$, the obtained feature map by the convolution operation can be shown as follows.

$$\begin{aligned} u_{i,j,k}^{(l_1)} &= (U^{(l_1-1)} * W_k^{(l_1)})(i,j) + w_{bk}^{(l_1)} \\ &= \sum_{p=1}^{P} \sum_{m=1}^{M'} \sum_{n=1}^{N'} w_{m,n,p} a_{i+m,j+n,p}^{(l-1)} + w_{bk}^{(l_1)} \end{aligned} \tag{5}$$

$$a_{i,j,k}^{(l_1)} = f(u_{i,j,k}^{(l_1)}) \tag{6}$$

where $f(\cdot)$ is the activation function and $a_{i,j,k}^{(l_1)}$ is the activated value of the unit in the i^{th} row and j^{th} column of the feature map. Therefore, $u_{i,j,k}^{(l_1)}$ is the value before activation. $w_{bk}^{(l_1)}$ denotes the bias of the k^{th} filter and is usually a single numeric value. $a_{i+m,j+n,d}^{(l_1-1)}$ is the activated value of unit in the $(i+m)^{th}$ row and $(j+n)^{th}$ column. Besides the convolution layers, the full connection layers are used to construct the basic training structure which is similar to DBA. Then, the similar forward propagation and back propagation processes are repeated to fine-tune the whole CNN structure.

IV. Proposed Deep Learning Based Traffic load prediction

In SDN-IoT, the central controller is the brain to control all functions of switches. All control and computation tasks are handled in the control controller, which is a totally centralized control system. In order to research the performance between using centralized SDN system and simi-centralized or distributed conventional control system without centralized SDN, we separately design our deep learning based traffic load prediction algorithm into three different systems.

In the conventional network, the switch (i.e., router, in order to easily describe, we still simply call a router in conventional network a switch) only knows local information and communication with each other in a distributed manner, which is referred to as a distributed control system. There is also a kind of mixed control system, in which the central controller is deployed with limited computation and communication ability. The limited central controller only knows part of the global information. In such a mixed system, the switches need to handle a part of the tasks in a localized manner and suffer from limited service from the central controller. Such a system can be treated as a semi-central control system. Based on these three different control systems, we propose three deep learning based traffic load prediction methods, namely, Central control based Traffic load Prediction (CTP), Semi-Central control Traffic load Prediction(S-CTP), and Distributed control Traffic load Prediction (DTP). Next, we describe the three prediction methods, respectively.

It is worth mentioning that apart from the link condition of each switch, the main factors influencing the traffic load is the arrival traffic flow. As mentioned ealier, in each switch, the traffic load sequence, TL, consists of two parts: the relayed traffic flow from other switches denoted by TL_rel, and the integrated traffic flow composed by the sensing data from devices in the sensing plane denoted by TL_int. Therefore, $TL = TL\_rel + TL\_int$.

A. Traffic Load Prediction in Central Control System:CTP

In the prediction process of central control system, there are four phases, i.e., data collection phase, training phase, prediction phase, and online training phase.

*1) **Data Collection Phase**:* In the central control system, all the information of switches are periodically collected by the central controller. The central controller records the traffic load sequence, TL, of every switch in the last N time slots. And the length of each time slot is represented as Δ. The traffic load of switch i in last time slot k is recorded as tl_k^i. Then, the past traffic loads TL^i of switch i are formed as a length-N vector, $TL^i = \{tl_k^i, tl_{k-1}^i, \cdots, tl_{k-N+1}^i\}$, where N represents the number of considered past time slots. N depends on the complexity of input data and is decided according to the training performance. In this case, the controller collects all traffic load series of every switch, and formats

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal

them as a traffic load matrix $TL = \{TL^1, TL^2, \cdots, TL^M\}$. From the point of the time series, the traffic loads of all switches in the last N time slots can be also represented as $TL = \{tl_k, tl_{k-1}, \cdots, tl_{k-N+1}\}$.

After data collection, the traffic load matrix TL is used as the input of training data. In the next time slot, the central controller records the traffic load as the real future traffic load $tl_{k+1} = \{tl^1_{k+1}, tl^2_{k+1}, \cdots, tl^M_{k+1}\}$ which will be utilized as the output of training data. After thousands of time slots, the central controller collects thousands of such labeled data and adopt those labeled real data to train the deep neural network in the training phase.

*2) **Training Phase**:* In this phase, in order to obtain a better training performance, we use a Deep Convolutional Neural Network (deep-CNN) to fit our matrix based training data [27]. In our earlier work [24], we compared the training performance with different output formats, and the result shows that the complex output significantly impair the training accuracy. In other words, utilizing only one deep-CNN to predict the future traffic load of all switches, which needs to use the full tl_{k+1} as the output, is too resource-consuming and has a significantly low accuracy. Therefore, we decouple the complex of output and use M deep-CNNs, where each deep-CNN is only used to predict the traffic load of one switch. Thus, the central controller only uses the future traffic load of one switch as the output of corresponding deep-CNN. For example, the training data of deep-CNN CNN^i is $(x_{input}, y_{output}) = (TL, tl^i_{k+1})$. Then, the central controller trains all the deep-CNNs, respectively, to obtain all the stable weight matrices.

*3) **Prediction and Accuracy Calculation Phase**:* In the prediction phase, the central controller undertakes the future traffic load prediction and calculates the prediction accuracy. In this phase, the weight matrix of each deep-CNN obtained in the training phase is adopted to predict the future traffic load, which is a forward propagation process as mentioned in Sec. III-C. The output of all deep-CNNs is recorded as $TLP_{k+1} = \{tlp^1_{k+1}, tlp^2_{k+1}, \cdots, tlp^M_{k+1}\}$. As mentioned above, the real future traffic load of time slot $(k+1)$ is recorded as $TL_{k+1} = \{tl^1_{k+1}, tl^2_{k+1}, \cdots, tl^M_{k+1}\}$. Therefore, we can calculate the prediction accuracy according to the following equation.

$$\frac{1}{K \times M} \sum_{k=0}^{K-1} \sum_{i=1}^{M} \frac{|tlp^i_{k+1} - tl^i_{k+1}|}{tl^i_{max}}, \tag{7}$$

where, K represents the total number of considered time slots. tl^i_{max} represents the maximum traffic loads of switch i, Here, we simply consider the maximum traffic load is equal to the maximum buffer size of the switch.

*4) **Online Training Phase**:* If the generation policy of input traffic always acts as a certain pattern, the training and prediction processes, based on only the existing training data, are reasonable. However, in a practical network, the generation policy of the input traffic may change because of some reasons, such as some devices break down, or some new sensing tasks are assigned to existing devices. Based on such situations, the training process should also be adapted correspondingly. Then the online training phase is necessary for adjusting the deep-CNNs to adapt to the new environment.

In this online training phase, each switch continuously records the traffic load data, and the training phase is processed periodically with the collected new training data. Therefore, the weight matrices are periodically adjusted.

B. Traffic Load Prediction in Semi-central Control System:S-CTP

In this kind of system, we consider the central controller only has some limited computation ability and the switches need to finish some tasks in a localized manner. In this case, each switch makes some simple pre-prediction just with the local information to alleviate the computational burden of central controller. And the final prediction is still conducted by the central controller with integrated global pre-prediction information from all switches.

*1) **Data Collection Phase**:* In the central control system, the central controller predicts the traffic load based on collected traffic patterns of all switches, that needs highly central computation ability and correspondingly fast communication mechanism of SDN technique. However, with the limited ability of the central controller in a semi-central control system, each switch cannot simply transfer all raw traffic information to the central controller because this will put much burden on the central controller. Thus, in the semi-central control system, switches should perform some pre-treatment of the raw data and send less information to the central controller to decrease both computation and signaling overheads of the central controller. In this case, for each switch i, it records the traffic load tl^i_k of the last time slot, and also separately records the relayed traffic load TL_rel^i and integrated traffic load TL_int^i of the last N time slots. Then, each switch i predicts the future integrated traffic load $tlp\_int^i_{k+1}$ of the next time slot by using recorded TL_int^i as input. This training and prediction process is conducted in the training phase. Then, the switch sends the obtained $tlp\_int^i_{k+1}$ and recorded traffic load of last time slot tl^i_k to the central controller. The central controller collects the data from all switches, and constructs them as the training data tl_k and tl_int_k.

*2) **Training Phase**:* The training phase consists of two steps. The first step is that each switch trains a local neural networks to predict its future integrated traffic load with its past N-time-slot integrated traffic loads. Therefore, for switch i, the training data of its local neural network can be represented as $(x_{input}, y_{output}) = (TL\_int^i, tl\_int^i_{k+1})$. Since the input is much simpler compared with the input of deep-CNN utilized in the central control system and the training can be treated as the function fitting process between the input and output, here, we can just use DBN mentioned in Sec. III-C to perform this training process. As we mentioned in the data collection phase, the trained DBA will be utilized to predict the future integrated traffic load which is represented as $tlp\_int^i_{k+1}$, and the results will be periodically sent to the central controller.

When switches finish self prediction and send the result to the central controller, the central controller performs the final prediction with last time slot traffic load

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal

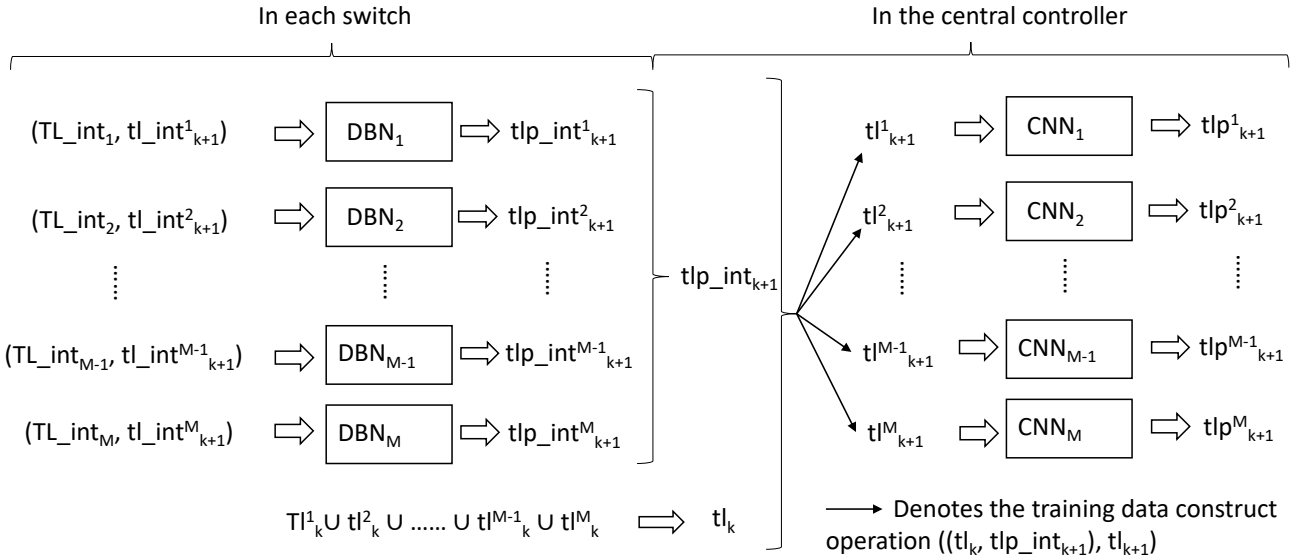


Fig. 3: The training and prediction phase in a semi-central control system.

tl_k and predicted integrated traffic load tlp_int_{k+1} of all switches. Since the traffic load and integrated traffic are two different network features, we form them as two channels of the input data, similar to our earlier research in [27]. Therefore, the input of training data can be formed as a matrix $(tl_k, tlp\_int_{k+1}) = (\{tl_k^1, tl_k^2, \cdots, tl_k^M\}, \{tlp\_int_{k+1}^1, tlp\_int_{k+1}^2, \cdots, tlp\_int_{k+1}^M\})$. As mentioned earlier, the deep learning structures in the central controller are utilized to predict the future traffic loads of all switches. Similar to the central control based prediction, we utilize M deep-CNNs to make the prediction to alleviate the computational burden and guarantee the accuracy. Therefore, for CNN^i, its labeled training data is formed as $(x_{input}, y_{output}) = ((tl_k, tlp\_int_{k+1}), tl_{k+1}^i)$.

Except the above-mentioned two phases, the prediction phase and online phase in the semi-central control system are almost the same as those in the central control system. The whole training process of each switch and the central controller is shown in Fig.3.

C. Traffic Load Prediction in Distributed Control System:DTP

In the conventional distributed network, the switches (i.e., router) do not know the global information, and the prediction must be executed in each switch only according to its local information. Thus, a local information based distributed traffic load prediction method is designed as follows.

In the distributed control system, each switch only collects its own traffic load including the relayed traffic load and integrated traffic load. Without additional information of other switches, the relationship between two kinds of traffic loads in different time slots becomes more complex. Therefore, the deep-CNNs utilized in this system are much wider and deeper than the deep-CNNs used in the central and semi-central control systems.

In order to get better training performance, we try two forms of the training data. The first one is to separate the integrated traffic load and relayed traffic load as input. Therefore, switch i records the integrated traffic load TL_int^i and relayed traffic load TL_rel^i of the last N time slots. Then, the two traffic loads are constructed to a two channel matrix as input of training data. Correspondingly, the traffic load in the next time slot, tl_{k+1}^i, is taken as the output. Thus, the training data can be represented as $(x_{input}, y_{output}) = ((TL\_rel^i, TL\_int^i), tl_{k+1}^i)$.

The second kind of input is only the combined traffic load TL^i. In this case, the training data can be denoted as $(x_{input}, y_{output}) = (TL^i, tl_{k+1}^i)$. And in the simulation (presented later in sec. VI), we compare the two kinds of training data and find that in current simulated network environment, both methods can achieve the same accuracy (i.e., above 85% in the network with 16 switches.). However, it takes more time for the first method to converge. Thus, we temporarily use the second method as the DTP training data in our research.

The CTP, SCTP and DTP methods are designed to fit the aforementioned three different kinds of control systems. The comparison of the prediction performance with those different methods are researched in section. VI.

V. Proposed Deep Learning Based Partially Channel Assignment

After the traffic loads of the next time slot are predicted by our proposed prediction methods, many existing channel assignment algorithms which are based on the traffic profile can be used to assign proper channel to each link. However, due to the problem we mentioned in Sec. I, the conventional channel assignment algorithms cannot meet the new requirement of the future SDN-IoT. Aided by the high computation ability in the future SDN [26], we propose a new deep learning based channel assignment algorithm, which shows better convergence performance than the conventional algorithms, and leads to better network throughput.

In our proposal, we use deep learning to train the network with the data from our prior research [14], in which, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal

partially overlapping channels are assigned to each link by using an anti-coordination game. In the anti-coordination game based partially channel assignment algorithm (AC-POCA), each router (i.e., switch) chooses the channel of its links by using a utility function and plays game with other switches. Different from cooperative game, AC-POCA can always get a unique stable state in the network, and such uniqueness of AC-POCA makes the algorithm appropriate to be trained by deep learning and gets almost the same accuracy as that of AC-POCA. If we set the AC-POCA as a benchmark, the deep learning based channel assignment algorithm can get 100% accuracy compared with the benchmark. Besides the same channel assignment accuracy, the deep learning based assignment algorithm can save the game process time between switches, leading to much faster convergence.

TABLE II: Learning performance with different combination of features

	HC	IF	HC+IF	TL	HC+IF+TL
Accuracy	23%	24%	67%	100%	100%
Epochs	∞	∞	∞	500	800

To train the network, we try to find the main features of training data. In ACPOCA with fixed topology, traffic load is the main feature to order the routers in the queue of game, and the order significantly affects the channel assignment result of each router. In the intuition of human, the traffic load (TL) should be the main feature of training data, and some other features should also be considered such as the hop count (HC) to the gateway and interference factor (IF) of each link. Then, we respectively use those features and some combinations of them to construct the different format of input of training data. In the experiment result shown in Table II, we can find that any feature combination containing the feature of traffic load can get 100% accuracy. To the opposite extreme, the accuracy of using any combination without traffic load is less than 70%. Besides the accuracy rate of different combinations, we compare the training epochs (i.e., training time) of the combinations containing traffic load. And the result shows that the method only using traffic load as input of training data can get the best performance of training. Therefore, in the training process, the traffic load is utilized as the main feature to construct the input of training data.

Here, we divide the channel assignment algorithm into two parts. In the first part, we propose a Deep Learning based Partially Overlapping Channel Assignment algorithm (DLPOCA). In the second part, we further propose an intelligent deep learning channel assignment strategy which joints the DLPOCA with traffic load prediction algorithm, referred to as TP-DLPOCA, to obtain further improved performance.

A. Deep Learning based Channel Assignment

To better describe our proposal, we divide the whole assignment process into two steps, i.e., the training phase and dynamic channel assignment phase.

1) Training Phase: Here, we use the traffic load and channel assignment result of AC-POCA as the training data set. Before using the data set, we need to characterize the training data into a suitable format. As we described above, we use the traffic load as the main feature to construct the input of training data. Such training data format is denoted as $tl_k = \{tl_k^1, tl_k^2, \cdots, tl_k^M\}$. Because the AC-POCA only considers the current traffic load, and the result is not affected by the traffic load of the past time sequence, we only use the traffic load of the last one slot as the input of training data.

Then, we consider that the assigned number of each link is recorded as the output of training data. If the scale of the network is significantly large, the number of links is large to make the output very complex. As mentioned in our previous work [24], the complex output will significantly decrease the training accuracy. Therefore, as the same method employed in our traffic prediction algorithm, we use $M \times E_{max}$ neural network to separately train the network, where E_{max} denotes the maximum number of active links of each node. For example, for 802.11 2.4 GHz links, because of self-interference, the same channel cannot be assigned to two links of one node. Then, the maximum number of active links E_{max} is 11, which is equal to the maximum number of channels C_{max}. Since each neural network is only used to predict the channel for one link, the number of total neural networks is equal to the number of links. And the neural network corresponding to the j^{th} link in switch j is recorded as $\{NN_{i,j} | i \leq M, j \leq E_{max}\}$. For each neural network, the output is characterized as a vector consisting of C_{max} binary elements, which can be denoted as $L = \{l_1, l_2, \cdots, l_{C_{max}-1}, l_{C_{max}} | l \in \{0, 1\}\}$. And if channel i is assigned, the value of the i^{th} element is 1, otherwise 0. Therefore, the training data of each neural network is indicated as $(x_{input}, y_{output}) = (tl_k, L)$.

With the training data, we try different kinds of neural network structures and different parameters for training. The comparison of the training results of different structures and parameters is shown in Sec. VI-B. Because of the large number of neural network and data set, this training process is better to be processed in the central controller. And the bias and weight matrices of all neural networks are recorded and updated in the central controller. The trained weight matrix of each switch is recorded as $\{WM_{i,j} | i \leq M, j \leq E_{max}\}$.

2) Dynamic Channel Assignment Phase: After the training process, the central controller sends the copy of the trained weight matrices to each corresponding switch. Each switch only stores the weight matrices corresponding to its own links. Then, during the packet transmission period, the central controller sends the traffic load information tl_k to each switch periodically, and each switch uses the current traffic load information as the input to trigger a forward propagation process with the corresponding weight matrix to get the output L^{k+1} (i.e., the binary vector of chosen channel). If the already assigned channel L^k of the link is different from the new one L^{k+1}, the switch confirms the new channel number with the other switch on the other side of the link. If both switches get the same result, they change the channel of this link to the new one. Otherwise, the switches report the different results to the central controller. The whole process is shown in Alg. 1.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal

	PIC	EIC
R	100	100
Q	30	30
W	10	20
n_p	80	50
n_e	20	50

TABLE III: The configuration of PIC and EIC

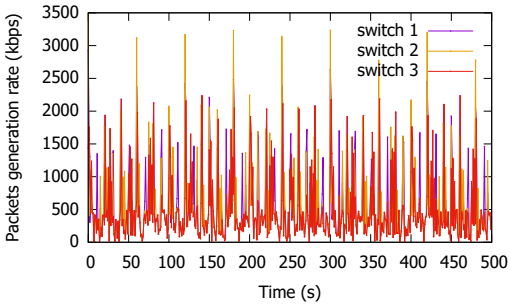


Fig. 4: The integrated traffic pattern of different switches in the periodic intensive case (PIC).

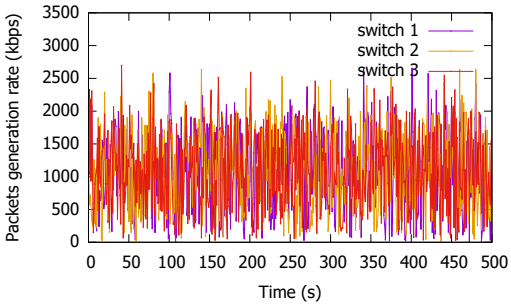


Fig. 5: The integrated traffic pattern of different switches in the event intensive case (EIC).

Algorithm 1 Algorithm of DLPOCA

Input: Trained weight matrices $\{WM_{i,j}|i \leq M, j \leq E_{max}\}$; each switch $\{s_i|i = 1 \rightarrow M\}$, traffic load tl_k.
Output: The assigned channel L_j^{k+1} of each link $\{e_j|j = 1 \rightarrow E_{max}\}$.

for $j = 1$ to E_{max} **do**
  The switch on the other side of link e_j is record as $\{s_f|f \leq M\}$.
  s_i use tl_k as input to trigger forward propagation with $WM_{i,j}$, the result is recorded as L_j^{k+1}.
  if $L_j^{k+1} \neq L_j^k$ **then**
    if $L_j^{k+1} = L_f^{k+1}$ of s_f **then**
      Assign L_j^{k+1} to link e_j.
    else
      Feedback the wrong information to central controller.
    end if
  end if
end for

Algorithm 2 Algorithm of TP-DLPOCA

Input: Trained weight matrices $\{WM_{i,j}|i \leq M, j \leq E_{max}\}$; each switch $\{s_i|i = 1 \rightarrow M\}$.
Output: The predicted traffic load tlp_{k+1}; assigned channel L_j^{k+1} of each link $\{e_j|j = 1 \rightarrow E_{max}\}$.

if Prediction model = CTP **then**
  s_i send TL^i to central controller.
  The central controller collects all $\{TL^i|i \rightarrow M\}$, and execute CTP prediction algorithm to get tlp_{k+1}.
  Central controller sends tlp_{k+1} to all switches.
else if Prediction model = S-CTP **then**
  s_i uses $TL^i\_int^i$ as input to calculate $tlp\_int_{k+1}^i$
  s_i sends $tlp\_int_{k+1}^i$ to central controller.
  The central controller collects all $\{tlp\_int_{k+1}^i|i \rightarrow M\}$ and $\{tl_k^i|i \rightarrow M\}$, then executes S-CTP prediction algorithm to get tlp_{k+1}.
  Central controller sends tlp_{k+1} to all switches.
else if Prediction model = DTP **then**
  s_i uses TL_rel^i and TL_int^i as input to calculate tlp_{k+1}^i by executing DTP prediction algorithm.
  s_i sends tlp_{k+1}^i to central controller.
  The central controller collects all $\{tlp_{k+1}^i|i \rightarrow M\}$ to construct as tlp_{k+1}.
  Central controller sends tlp_{k+1} to all switches.
end if
tlp_{k+1} is used as input to execute algorithm. 1.

With deep learning based channel assignment, the channel assignment result can be simply obtained via a forward propagation process, which is much faster than the the game theory based channel assignment and saves most of the communication cost/signaling overhead. This is because in the conventional game theory based channel assignment methods, each router needs to keep receiving and updating the channel statements of all other routers in every iteration. Consequently, the more iterations of decision process, the heavier signaling overhead. On the other hand, in our proposed deep learning based channel assignment, only one iteration is needed, which is the main reason why our proposal can significantly outperform the conventional one.

B. Deep Learning based Channel Assignment jointed with Prediction

The traditional channel assignment performs the channel assignment according to the current (i.e., the last time slot) traffic load. This assumption is reasonable when the traffic load is constantly and slowly changed. However, in the practical environment, the traffic load is not so smooth. And in different applications, the traffic load may suddenly change or have more complex features. To solve this problem, we further combine the deep learning based channel assignment with the proposed traffic load prediction method, which is named as TP-DLPOCA. To compare it with DLPOCA, we replace the input traffic load in the training data with the predicted traffic load obtained via the deep learning structures described in Sec. IV. Since, the traffic load of all switches in next slot is predicted and formated as $tlp_{k+1} = \{tlp_{k+1}^1, tlp_{k+1}^2, \cdots, tlp_{k+1}^M\}$, the training data are denoted as $(x_{input}, y_{output}) = (tlp_{k+1}, L)$.

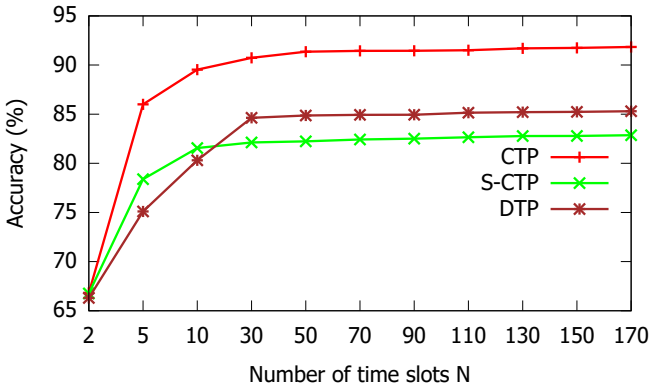
Except the new training data set with the predicted traffic load, the training phase of TP-DLPOCA is the same as the DLPOCA. In the dynamic channel assignment phase, the traffic load prediction is done before the channel assignment. Then, the central controller sends the predicted traffic load information tlp_{k+1} to every switch. The entired process is shown in Alg. 2.

VI. Performance Evaluation

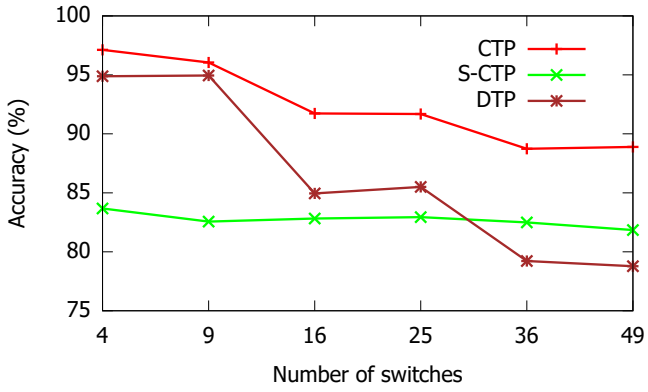
In this section, we evaluate our proposal from three aspects: the prediction accuracy, the performance of DLPOCA, and the performance of TP-DLPOCA.

We simulate the scenarios with the configuration using C++/WILL [32] API as follow. A square area is set with same maximum width and length which is proportional to the number of switches and devices in the network. All switches in the network are randomly deployed in this area. As we described in Sec.III-A, there are different kinds of devices

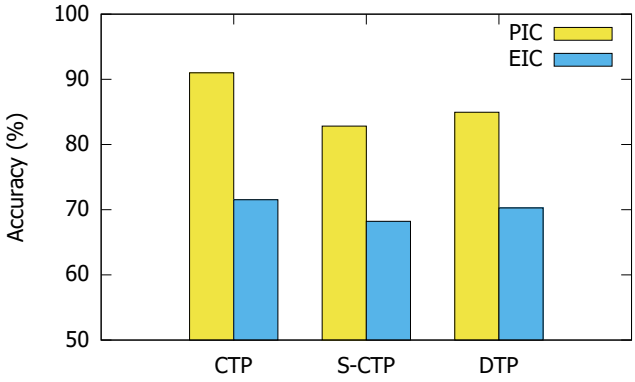
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal



(a) The prediction accuracy with different numbers of time slots, N



(b) The prediction accuracy with different numbers of switches



(c) The prediction accuracy with different kinds of data generation method

Fig. 6: The performance comparison of the three kinds of proposed mechanisms.

deployed in the control area of each switch. In the conducted simulations, we set the average number of devices belong to each switch area $R = 100$ and the kinds of periodic sensing devices $Q = 30$. In the beginning of simulation, we randomly choose 10 out of the Q (i.e., 30) kinds of periodic sensing devices to be deployed in the network. Consider the number of periodic sensing devices in each switch control area denoted as n_p and the number of event driven devices denoted as n_e. For simulating the influence of different kinds of devices, we considered two cases with different ratios of periodical and event driven devices deployed in the sensing plane. In one case, we deploy 80 periodic sensing devices and 20 randomly event driven sensing devices to each switch control area (i.e., $n_p = 80$ and $n_e = 20$), which represents the high ratio of periodic sensing devices, we briefly call it Periodic Intensive Case (PIC). In the other case, in each switch control area, we only deploy 50 periodic sensing devices and increase the number of randomly event driven sensing devices to 50, which represents the case with high ratio of event driven devices and briefly named as Event Intensive Case (EIC) in our research. We set the kinds of event driven sensing devices $W = 10$ in PIC and $W = 20$ in EIC. The data collection policy of event sensing devices is random. The detailed configuration and one example of integrated traffic load of different switches in the two different cases are shown in Table. III, Fig. 4 and Fig. 5. Consider the distance between switches is randomly set ranging from 10m to the maximum width of the square place. The gateway is positioned at the top right corner in the simulated network that is the farthest from the user devices. To simplify the simulation, we use the similar spectrum configuration of [14], the multi-channels, multi-radios are assumed to be equipped on each switch and is operated with IEEE 802.11g wireless technology. The interference model of those wireless channels is mentioned in Sec. III, and the $IR(\sigma)$ is shown in Table I. The data rate of each link is set to 8Mbps. We conduct our network with the number of switches from 9 to 64 to show the different performances in various network environments.

A. Prediction Accuracy

At first, we evaluate the accuracy performance of traffic load prediction with three proposed mechanisms, namely, CTP, S-CTP, and DTP. In the case of PIC, the number of switches and the slot length are 16 and 1s, respectively. We compare the prediction accuracy of three mechanisms with different number of slots N, which is an important parameter of the prediction algorithm mentioned in Sec. IV-A. In Fig. 6(a), we can notice that the accuracy of all three mechanisms increases with the increasing value of N before $N = 30$. When N exceeds 30, the accuracy slightly increases and intends to be stable. That indicate that $N = 30$ is the threshold, which represents whether the features used in the input data are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal

enough for training. Furthermore, the figure shows that, when N is below the threshold, the accuracy of S-CTP is better than that of DTP, while the accuracy of DTP is higher when N is above the threshold. And the accuracy with CTP is always better than S-CTP and DTP (more than 90%).

With the different kinds of policies chosen by devices, the features of traffic patterns become more complex. Thus, choosing the suitable time slot Δ to fit the features of traffic pattern is very important to increase the prediction accuracy. In addition, we compare the accuracy of the three mechanisms with different lengths of Δ. This simulation is conducted in the situation of PIC, and the number of switches and the value of N are 16 and 70, respectively. The result indicates that the prediction accuracy is significantly affected by Δ. For all three mechanism, there are two crests (i.e. $\Delta = 4$ *or* 7) which show the most suitable slot length Δ for our traffic patterns. However, regardless of the Δ value we choose, the accuracy of CTP is always better than the other two mechanisms.

Fig. 6(b) demonstrates the accuracy of the three mechanisms with different numbers of switches. This simulation is conducted in the situation of PIC and $N = 70, \Delta = 1s$. The switches are deployed as described in Sec. III-A. From the figure, we can notice that the prediction accuracies of CTP and DTP decrease with an increasing number of switches. However, S-CTP always exhibits a stable prediction accuracy. And when the number of switches is more than 25, the performance of S-CTP is even better than that of DTP. When the number of switches exceeds 25, the prediction accuracy of CTP also tends to be stable and can achieve nearly 90% accuracy. This means that the proposed deep learning based prediction algorithm is also suitable for a large scale network.

Furthermore, as shown in Fig. 6(c), we compare the three kinds of deep learning based traffic load prediction accuracy in situations of PIC and EIC. The prediction accuracies in PIC always significantly outperforms that in EIC. This is because there are so many random events in EIC and it is hard to track its policy. On the other hand, considering the fact that the event driven still has some rules in practical networks, such situation can be further researched in the future works.

Thus, the results show the advantage of using the SDN central control system. This is because the high computation ability and communication mechanism in SDN allows more complex information to be used as training data in learning process. Next, we investigate the performance of deep learning based channel assignment in SDN-IoT.

B. Performance of Deep Learning Based Channel Assignment

In this part, we compare the learning performance of POC with different learning structures and different learning parameters. Then, we compare the POC accuracy of our proposal. Finally, we compare the throughput between our proposed DLPOCA and traditional channel assignment algorithms (i.e., the orthogonal channel assignment, POC, AC-POCA).

In Fig. 7, we compare the training accuracy with different learning structures, i.e., Deep Belief Network (DBN) with 2 and 3 hidden layers. The number of nodes in each layer is set to 20 and 100. Here, we briefly call them 20-2-DBN, 20-3-DBN, 100-2-DBN, and 100-3-DBN, respectively. Then, we change the DBN structure into deep CNN with 1 and 2 convolution layers and 2 full connection layers, respectively. In the CNN, we set the size of convolution layer as 3×3, the number of nodes in full connection layer is 100, the number of channels in convolution layer is 20, and the padding and stride are set to 1. Correspondingly, we briefly call them 1-CNN, 2-CNN. Then, we compare those different training structures in different network structures. After running all those training processes with mini-batch size of 20 and 500 epoches, the accuracy result is shown in Fig. 7. From the result, we can notice that the accuracy is deeply related to the training structure, and the deep CNN is much better than DBN in our scenario. Moreover, the 2-CNN can always get 100% accuracy in our network and is chosen as our final training structure.

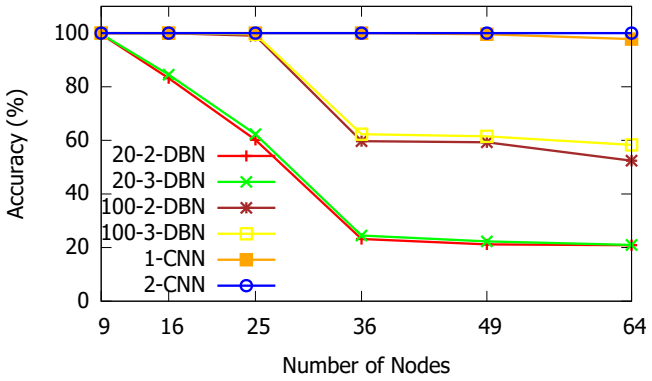


Fig. 7: The accuracy with different configuration of learning structure.

Our proposed DLPOCA chooses the deep-CNN as the training structure, and the simulation result demonstrates almost 100% accuracy. Then, we compare the convergence time (i.e., iteration times) of DLPOCA and conventional algorithms, Cooperative Channel Assignment Game (CoCAG) with Best Response (BR) and Smoothed Better Response (SBR) [11] and ACPOCA [14]. For ease of representation, the three methods are briefly referred to as CoCAG-BR, CoCAG-SBR and ACPOCA, respectively. We run the channel assignment process 100 times with randomly deployment of nodes and obtain the average number of iteration times.

Fig. 8 shows the comparison result of the convergence time. As described in Sec. V-A, with our proposed DLPOCA, the number of iteration times is always 1, which significantly outperforms conventional algorithms. In conventional algorithms, the switch chooses the channel of its links depends on the decisions of other switches. This means that the switches must wait until other prior switches finished their channel assignment. The more the iteration times, the longer time each switch needs to spend in channel assignment. This causes redundant convergence time.

Because of the redundant convergence time, redundant signaling correspondingly increases. During the convergence time, all links are down because of the channel reassignment, and the throughput decreases with such redundant convergence

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal

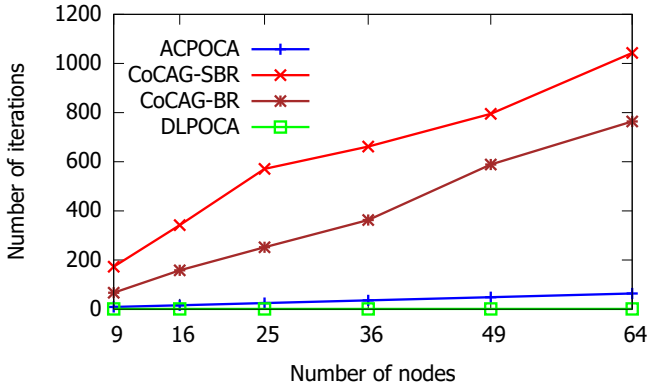


Fig. 8: The convergence compared with proposal and conventional algorithm.

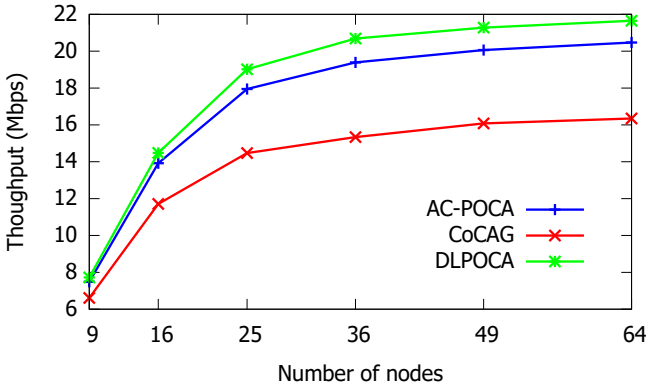


Fig. 9: The throughput compared with proposed DLPOCA and conventional algorithms.

time. However, with our proposed DLPOCA, the number of iteration time is always 1, and both the convergence time and signaling overhead are significantly low.

Considering the dynamics of network, the traffic load and link condition may change frequently, leading to the frequent channel reassignments. Here, we consider the traffic load in the situation of PIC. In order to simulate such a situation, we set the frequency of channel assignment in our simulation as 10s/1 (i.e., every 10 seconds execute once). During the channel assignment, all data transmissions are paused. And then, we compare the throughput between DLPOCA, ACPOCA and CoCAG. In this simulation, the buffer of each node is set to 100KB, and the packets are randomly discarded when the buffer is full. In order to show the advantage of our proposal more clearly, in this section, we just consider that all packets are generated normally, i.e., not in the situation of PIC or EIC. The packet generation rate of each node is set as 1Mbps. The packet size and signaling size both are set to 1kb. Then, we run the simulation over 1000s, and the throughput result is demenstrated in Fig. 9. From the above result, it can be noticed that the throughput of the proposed DL-POCA is always better than that of conventional channel assignment algorithms. This is because of the quick convergence of our proposal. The channel assignment of all routers can be decided only with one broadcast and finished almost immediately. On the contrary, in the conventional channel assignment algorithms, the switches need to make decisions one by one that causes a high number of iterations to converge. Especially by randomly Smoothed Better Response based game theoretical CoCAG, switch have to make many redundant decisions to maximize their utility.

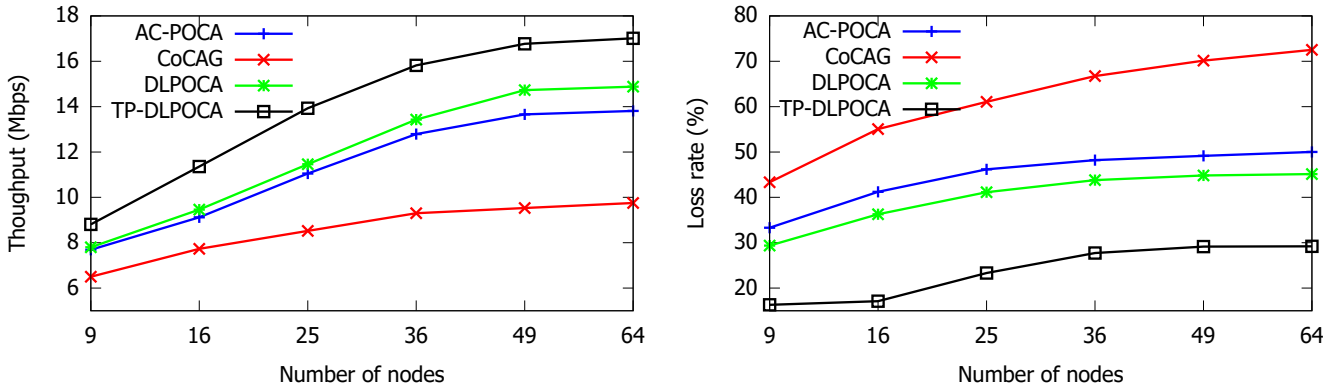
C. Performance of the Joint Deep Learning Based Prediction and Channel Assignment

To further improve the channel performance, we combine the intelligent deep learning based channel assignment with deep learning based traffic load prediction. The result of Fig. 10 shows the performance of our proposed TP-DLPOCA with conventional channel assignment algorithms (i.e., CoCAG, AC-POCA). In this simulation, the parameters are considered to be the same as those in Sec. VI-B. Figs. 10(a) and 10(b) demonstrate the comparison of throughput and packet loss rate of TP-DLPOCA and conventional algorithms, respectively. From the results, we can notice that the performance of the proposed TP-DLPOCA is much better than that of conventional algorithms and even DLPOCA. Furthermore, when the number of switches increases, the packets loss rate of TP-DLPOCA almost tends to be stable, indicating the advantage of TP-DLPOCA compared with conventional algorithms in a larger network. This is because TP-DLPOCA can predict the congestion and give the links in the congestion area higher priority to be assigned with high quality channels. And even when the network situation changes, the proposed TP-DLPOCA can still learn from the new situation and predict the congestion to assign suitable channels. This good performance of the proposed TP-DLPOCA can be credited as the online intelligent channel assignment strategy.

VII. Conclusion

The explosive growth of sensing data and quick response requirements of the IoT have recently led to the high speed transmissions in the wireless IoT to emerge as a critical issue. Assigning suitable channels in wireless IoT is a basic guarantee of high speed transmission. However, the conventional fixed channel assignment algorithms are not suitable in the IoT due to the highly dynamic traffic loads. Recently, the Software Defined Networking based IoT (SDN-IoT) is proposed to improve the transmission quality. Moreover, the deep learning technique has been widely researched in high computational SDN. Therefore, we first proposed a novel deep learning based traffic load prediction method to predict the future traffic load and network congestion. Then, a deep learning based partially channel assignment algorithm (DPPOCA) was proposed to intelligently assign channels to each link in SDN-IoT. Finally, we combine the deep learning based traffic prediction and channel assignment to propose a novel intelligent channel assignment algorithm (TP-DLPOCA), which can intelligently avoid traffic congestion and quickly assign suitable channels to the wireless links of SDN-IoT. Extensive simulation results demonstrate that our proposal significantly outperforms the conventional channel assignment algorithms.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2018.2838574, IEEE Internet of Things Journal



(a) The throughput compared with TP-DLPOCA and conventional algorithms(b) The packets loss rate compared with TP-DLPOCA and conventional algorithms

Fig. 10: The network performance comparison of the TP-DLPOCA, DLPOCA and conventional algorithms in terms of throughput, packets loss rate in the situation of PIC.

REFERENCES

[1] Y. Kawamoto, N. Yamada, H. Nishiyama, N. Kato, Y. Shimizu, and Y. Zheng, "A feedback control-based crowd dynamics management in iot system," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1466–1476, Oct 2017.

[2] Y. Kawamoto, H. Nishiyama, N. Kato, Y. Shimizu, A. Takahara, and T. Jiang, "Effectively collecting data for the location-based authentication in internet of things," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1403–1411, Sept 2017.

[3] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama, and N. Kato, "A survey on network methodologies for real-time analytics of massive iot data and open research issues," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1457–1477, thirdquarter 2017.

[4] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2017.

[5] S. Al-Rubaye, E. Kadhum, Q. Ni, and A. Anpalagan, "Industrial internet of things driven by sdn platform for smart grid resiliency," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2017.

[6] K. Sood, S. Yu, and Y. Xiang, "Software-defined wireless networking opportunities and challenges for internet-of-things: A review," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 453–463, Aug 2016.

[7] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, "A software defined networking architecture for the internet-of-things," in *2014 IEEE Network Operations and Management Symposium (NOMS)*, May 2014, pp. 1–9.

[8] F. Lin, C. Chen, N. Zhang, X. Guan, and X. Shen, "Autonomous channel switching: Towards efficient spectrum sharing for industrial wireless sensor networks," *IEEE Internet of Things Journal*, vol. 3, no. 2, pp. 231–243, April 2016.

[9] R. Zhang, M. Wang, X. Shen, and L. L. Xie, "Probabilistic analysis on qos provisioning for internet of things in lte-a heterogeneous networks with partial spectrum usage," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 354–365, June 2016.

[10] N. Zhang, N. Cheng, N. Lu, H. Zhou, J. W. Mark, and X. S. Shen, "Risk-aware cooperative spectrum access for multi-channel cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 3, pp. 516–527, March 2014.

[11] P. B. F. Duarte, Z. M. Fadlullah, A. V. Vasilakos, and N. Kato, "On the partially overlapped channel assignment on wireless mesh network backbone: A game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 119–127, January 2012.

[12] Y. Liu, R. Venkatesan, and C. Li, "Channel assignment exploiting partially overlapping channels for wireless mesh networks," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, Nov 2009, pp. 1–5.

[13] V. Bukkapatanam, A. A. Franklin, and C. S. R. Murthy, "Using partially overlapped channels for end-to-end flow allocation and channel assignment in wireless mesh networks," in *2009 IEEE International Conference on Communications*, June 2009, pp. 1–6.

[14] F. Tang, Z. M. Fadlullah, N. Kato, F. One, and R. Miura, "Ac-poca: Anti-coordination game based partially overlapping channels assignment in combined uav and d2d based networks," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2017.

[15] CISCO, "The Internet of Things Reference Model White Paper," *"http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf"*, (accessed March 2018).

[16] F. Al-Turjman, "Information-centric framework for the internet of things (iot): Traffic modeling and optimization," *Future Generation Computer Systems*, vol. 80, pp. 63 – 75, 2018.

[17] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325 – 349, 2005.

[18] S. Tozlu, M. Senel, W. Mao, and A. Keshavarzian, "Wi-fi enabled sensors for internet of things: A practical approach," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 134–143, June 2012.

[19] K. Sood, S. Yu, Y. Xiang, and S. Peng, "Control layer resource management in sdn-iot networks using multi-objective constraint," in *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, June 2016, pp. 71–76.

[20] M. Ojo, D. Adami, and S. Giordano, "A sdn-iot architecture with nfv implementation," in *2016 IEEE Globecom Workshops (GC Wkshps)*, Dec 2016, pp. 1–6.

[21] B. Nguyen, N. Choi, M. Thottan, and J. V. der Merwe, "Simeca: Sdn-based iot mobile edge cloud architecture," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, May 2017, pp. 503–509.

[22] H. Skalli, S. Ghosh, S. K. Das, and L. Lenzini, "Channel assignment strategies for multiradio wireless mesh networks: Issues and solutions," *IEEE Communications Magazine*, vol. 45, no. 11, pp. 86–95, November 2007.

[23] P. B. F. Duarte, Z. M. Fadlullah, K. Hashimoto, and N. Kato, "Partially overlapped channel assignment on wireless mesh network backbone," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, Dec 2010, pp. 1–5.

[24] N. Kato, Z. M. Fadlullah, B. Mao, F. Tang, O. Akashi, T. Inoue, and K. Mizutani, "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," *IEEE Wireless Communications*, vol. 24, no. 3, pp. 146–153, 2017.

[25] Z. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2017.

[26] B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "Routing or computing? the paradigm shift towards intelligent computer network packet transmission based on deep learning," *IEEE Transactions on Computers*, vol. 66, no. 11, pp. 1946–1960, Nov 2017.

[27] F. Tang, B. Mao, Z. M. Fadlullah, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "On removing routing protocol from future wireless

networks: A real-time deep learning approach for intelligent traffic control," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 154–160, February 2018.

[28] M. A. Hoque, X. Hong, and F. Afroz, "Multiple radio channel assignement utilizing partially overlapped channels," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, Nov 2009, pp. 1–7.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99. [Online]. Available: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf

[30] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, 2014, p. 17461751.

[31] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 4489–4497.

[32] "WILL API," *https://scarsty.gitbooks.io/will/content/*, (accessed December 2017).

FengXiao Tang (S'16) received the B.E. degree in measurement and control technology and instrument from the Wuhan University of Technology, Wuhan, China, in 2012 and the M.S. degree in software engineering from the Central South University, Changsha, China, in 2015. Currently, he is pursuing the Ph.D. degree in the Graduate School of Information Sciences (GSIS) at Tohoku University, Sendai, Japan. His research interests are unmanned aerial vehicles system, game theory optimization and deep learning.

Zubair Md. Faddullah (M'11, SM'13) is serving as an Associate Professor at the GSIS, Tohoku University. His research interests are in the areas of 5G, social networks, smart grid, and network security. He was a recipient of the prestigious Dean's and President's awards from Tohoku University in March 2011 for his outstanding research contributions. He was also recipient of the IEEE Communications Society Asia Pacific Board Outstanding Young Researcher Award in 2015 and the NEC Tokin foundation award in 2016.

Bomin Mao (S'15) received the BSc degree in telecommunications engineering and M.S. degree in electronics and telecommunications engineering at Xidian University, China, in 2012 and 2015, respectively. Currently, he is pursuing Ph.D. degree at the Graduate School of Information Sciences, Tohoku University, Japan. His research interests are involving wireless networks, software defined networks, quality of service, particularly with applications of machine intelligence and deep learning.

Nei Kato (M'03, A'04, SM'05, F'13) is a full professor and the Director of Research Organization of Electrical Communication(ROEC), Tohoku University, Japan. He has been engaged in research on computer networking, wireless mobile communications, satellite communications, ad hoc & sensor & mesh networks, smart grid, IoT, Big Data, and pattern recognition. He has published more than 400 papers in prestigious peer-reviewed journals and conferences. He is the Vice-President(Member & Global Activities) of IEEE Communications Society(2018-2019), the Editor-in-Chief of IEEE Network Magazine(2015-2017), the Editor-in-Chief of IEEE Transactions on Vehicular Technology(2017-), the Associate Editor-in-Chief of IEEE Internet of Things Journal(2013-), and the Chair of IEEE Communications Society Sendai Chapter. He served as a Member-at-Large on the Board of Governors, IEEE Communications Society(2014-2016), a Vice Chair of Fellow Committee of IEEE Computer Society(2016), a member of IEEE Computer Society Award Committee(2015-2016) and IEEE Communications Society Award Committee(2015-2017). He has also served as the Chair of Satellite and Space Communications Technical Committee(2010-2012) and Ad Hoc & Sensor Networks Technical Committee(2014-2015) of IEEE Communications Society. His awards include Minoru Ishida Foundation Research Encouragement Prize(2003), Distinguished Contributions to Satellite Communications Award from the IEEE Communications Society, Satellite and Space Communications Technical Committee(2005), the FUNAI information Science Award(2007), the TELCOM System Technology Award from Foundation for Electrical Communications Diffusion(2008), the IEICE Network System Research Award(2009), the IEICE Satellite Communications Research Award(2011), the KDDI Foundation Excellent Research Award(2012), IEICE Communications Society Distinguished Service Award(2012), IEICE Communications Society Best Paper Award(2012), Distinguished Contributions to Disaster-resilient Networks R&D Award from Ministry of Internal Affairs and Communications, Japan(2014), Outstanding Service and Leadership Recognition Award 2016 from IEEE Communications Society Ad Hoc & Sensor Networks Technical Committee, Radio Achievements Award from Ministry of Internal Affairs and Communications, Japan (2016) and Best Paper Awards from IEEE ICC/GLOBECOM/WCNC/VTC. Nei Kato is a Distinguished Lecturer of IEEE Communications Society and Vehicular Technology Society. He is also a fellow of IEICE.