National Defence Défense nationale

DEFENCE RESEARCH AND DEVELOPMENT CANADA (DRDC)

RECHERCHE ET DÉVELOPPEMENT POUR LA DÉFENSE CANADA (RDDC)

# A Deep Reinforcement Learning-based Trust Management Scheme for Software-defined Vehicular Networks

Dajun Zhang
F. Richard Yu
Department of Systems Computer Engingeering
Carleton University, Ottawa, Ontario

Ruizhe Yang
School of Information and
Communication Engineering
Beijing University of Technology

Helen Tang
DRDC – Ottawa Research Centre

DIVANet'18: Proceedings of the 8th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications

Montréal, Québec, Canada
28 October–2 November
2018
7 Pages

Date of Publication from Ext Publisher: October 2018

## Defence Research and Development Canada

**External Literature (N)**
DRDC-RDDC-2022-N286
November 2022

Canada

**IMPORTANT INFORMATIVE STATEMENTS**

This document was reviewed for Controlled Goods by Defence Research and Development Canada using the Schedule to the *Defence Production Act*.

Disclaimer: This document is not published by the Editorial Office of Defence Research and Development Canada, an agency of the Department of National Defence of Canada but is to be catalogued in the Canadian Defence Information System (CANDIS), the national repository for Defence S&T documents. Her Majesty the Queen in Right of Canada (Department of National Defence) makes no representations or warranties, expressed or implied, of any kind whatsoever, and assumes no liability for the accuracy, reliability, completeness, currency or usefulness of any information, product, process or material included in this document. Nothing in this document should be interpreted as an endorsement for the specific use of any tool, technique or process examined in it. Any reliance on, or use of, any information, product, process or material included in this document is at the sole risk of the person so using it or relying on it. Canada does not assume any liability in respect of any damages or losses arising out of or in connection with the use of, or reliance on, any information, product, process or material included in this document.

# A Deep Reinforcement Learning-based Trust Management Scheme for Software-defined Vehicular Networks

Dajun Zhang, F.Richard Yu
Depart. of Systems Computer Eng.
Carleton University, Ottawa, ON, Canada
dajunzhang@cmail.carleton.ca, richard.yu@carleton.ca

Ruizhe Yang
School of Information and Communication Engineering
Beijing University of Technology
yangruizhe@bjut.edu.cn

Helen Tang
Defence Research and Development Canada
Ottawa, ON, Canada
Tang.HY@forces.ca

## ABSTRACT

Vehicular ad hoc networks (VANETs) have become a promising technology in intelligent transportation systems (ITS) with rising interest of expedient, safe, and high-efficient transportation. VANETs are vulnerable to malicious nodes and result in performance degradation because of dynamicity and infrastructure-less. In this paper, we propose a trust based dueling deep reinforcement learning approach (T-DDRL) for communication of connected vehicles, we deploy a dueling network architecture into a logically centralized controller of software-defined networking (SDN). Specifically, the SDN controller is used as an agent to learn the most trusted routing path by deep neural network (DNN) in VANETs, where the trust model is designed to evaluate neighbors' behaviour of forwarding routing information. Simulation results are presented to show the effectiveness of the proposed T-DDRL framework.

## CCS CONCEPTS

• **Networks** → **Routing protocols**;

## KEYWORDS

Vehicular ad hoc networks, Software-defined Networking, Dueling deep reinforcement learning, Trust

**ACM Reference Format:**
Dajun Zhang, F.Richard Yu, Ruizhe Yang, and Helen Tang. 2018. A Deep Reinforcement Learning-based Trust Management Scheme for Software-defined Vehicular Networks. In *DIVANet '18: 8th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, Oct. 28–Nov. 2, 2018, Montréal, Québec, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3272036.3272037

## 1 INTRODUCTION

A vehicular ad hoc network (VANET) is a type of mobile ad hoc networks (MANETs) in the vehicular environment. With the rising demand of convenient, safe, and efficient transportation, VANETs act as a vital role in *intelligent transportation systems* (ITS) [2, 22]. However, bad effects of malicious vehicles and inefficient network utilization are two main challenges in deployment of VANETs [19].

Researchers have proposed many security mechanisms in order to enhance the security of VANETs [4, 12, 14]. The authors of [11] propose a discrete event based threat driven authentication approach, in which the scheme aims to solve the security communications between vehicle-to-vehicle (V2V). A trust based framework is proposed in [3] that provides collaboration trust, behavioural trust and reference trust values for VANETs to estimate trust degree of each node. Wang *et al.* [16] introduce a field game model to solve the security problems in VANETs. Tangede *et al.* [15] present a a decentralized and scalable privacy-preserving authentication (DSPA) scheme for secure vehicular ad hoc networks (VANETs).

Although many researchers have already done some excellent works on VANET security issues [7, 9], they are still hard to ensure safety because most existing security works couple data forwarding with control. Software-defined networking (SDN) and virtualization [8, 20, 21] have become an emerging technology, which enables researchers to solve the above problems. Meanwhile, a lot of researchers proposed various schemes based on machine learning algorithms to solve the VANET challenges [5, 6]. The deep reinforcement learning (DRL) algorithm is introduced by [13], and makes big improvements compared with the traditional machine learning algorithms. Moreover, dueling network architecture for deep reinforcement learning is proposed by [17], and also makes a big improvement compared with the traditional DRL.

In this paper, we focus on applying the dueling network architecture for DRL and deploying centralized SDN controller into VANETs. Specifically, by decoupling the control and data forwarding plane in VANETs, we deploy the dueling DRL algorithm into a logically centralized controller. Therefore, our proposed scheme will have some features like high flexility, self-learning capability, and programmability.

The rest of paper is organized as follows: The background information of deep reinforcement learning is described in

*DIVANet '18, October 28–November 2, 2018, Montréal, Québec, Canada*

ACM ISBN 978-1-4503-5964-1/18/10. . . $15.00
https://doi.org/10.1145/3272036.3272037

Section II. Section III describes our system model, and T-DDRL approach is formulated in this section. The performance of T-DDRL is evaluated and comparison is described in Section IV. Finally, some conclusions are given in Section V.

## 2 BACKGROUND

In this section, we first introduce the basic idea of deep reinforcement learning, and then we illustrate the concept of dueling network architecture for deep reinforcement learning.

### 2.1 Deep Reinforcement Learning

A deep reinforcement learning concept was introduced by [13], and aims to solve the instability of traditional Q-network. There are two important improvements compared with traditional reinforcement learning method:*experience replay* and *target Q-network*. Experience replay stores trained data and then randomly samples from the pool. Therefore, it reduces the correlation of data and improves the performance compared with the previous reinforcement learning algorithms [13]. Meanwhile, *target Q-network* is another improvement in DRL method. That is, it calculate a target Q-value using a dedicated target Q-network, rather than directly using the pre-updated Q-network. The purpose of this is to reduce the relevance of the target calculation to the current value.

More precisely, each time after training for a period of time (i.e., every C steps), the parameters of the current Q-network are copied to the target Q-network. Such modification can make the learning procedure more stable than the previous Q-learning.

### 2.2 Beyond Deep Reinforcement Learning

Although deep reinforcement learning makes big improvements comparing with traditional machine learning, many researchers still make great efforts for even greater performance and higher stability. Here, we introduce a recent improvement: dueling deep Q-network(DDQN).

The core idea of DDQN is that it always not need to estimate the value of taking each available action. For some states, the choice of action makes no influence on these states themselves. Thus, the network architecture of DDQN can be divided into two main components: value function and advantage function. The value function is used to represent how good it is to be in a given state, and advantage function can measure the relative importance of a certain action compared with other actions. After value function and advantage function are separately computed, their results are combined back to the final layer to calculate the final Q-value. The mechanism of DDQN would lead to better policy evaluation comparing with DRL.

In our paper, we use DDQN to optimize the communication of connected vehicles, and the formulation process is described in the following section.

## 3 THE FRAMEWORK OF TRUST BASED OPTIMAL ROUTING IN VANETS

In this section, we formulate the optimization problem of VANET routing selection as a deep reinforcement learning process. Specifically, there are two main components in our proposed framework: path learning and trust computation. In these two phases, because of bad effects of malicious vehicles, each node in VANETs aims to learn the best routing path for communicating with each other. The decision problem for routing selection is that a vehicle in VANETs attempts to select a reliable neighbour vehicle as its next hop. The problem is under an assumption that routing selections of intermediate nodes in VANETs are stochastic, and depend on some routing information, i.e., trust information and each vehicle's location. In this section, first we introduce our system model, and then illustrate the method of T-DDRL.

### 3.1 System Model

Consider a twelve-vehicle VANET environment shown in Fig. 1, where vehicle $n_0$ aims to communicate with vehicle $n_{11}$. Each vehicle contains some important information, such as speed, direction and trust information, to build connection with its neighbors. Since the malicious vehicles will randomly drop the information, each vehicle in the environment aims to select the most trusted neighbor to transfer the routing information, and finally establishes a secure routing path to communicate with each other.

Fig. 2 shows the framework of our proposed T-DDRL method for VANET routing selection problem, where an SDN controller acts as an agent to interact with a VANET environment at discrete time steps, $t = 0, 1, 2, ...$, and the goal of the agent is to find an optimal secure routing path from the source vehicle to destination vehicle. As shown in fig.2, the network infrastructure is served as the environment, and the control plane of SDN performs an agent interacting with VANET network as the environment for learning task.

We define three-tuple $\{S, A, R\}$ for our proposed T-DDRL framework. $S$ denotes the set of states that the agent observes from the environment, and $A$ determines the possible action set of the agent. In time step $t$, SDN controller observes the environment and gets a state $s_t \in S$, then it takes an action $a_t \in A$ and gets a reward $r_t \in R$. Meanwhile, the agent aims to get long term rewards for each state-action pair. The agent keeps track of the Q-value for each state-action pair $Q(s_t, a_t)$.

We assume that there are $N = \{0,1,2,...,n\}$ vehicles in a VANET environment. A node $n_0 \in N$ in the environment is served as the source vehicle that sends new routing information to the network in each time step $t$. We assume all the information have the same destination $n$ at time $t$ for notational convenience. The T-DDRL approach can be used in the path discovery and packet forwarding logic. Next, we define the system state $s_t$, agent action $a_t$, and system reward $R_t$, respectively.

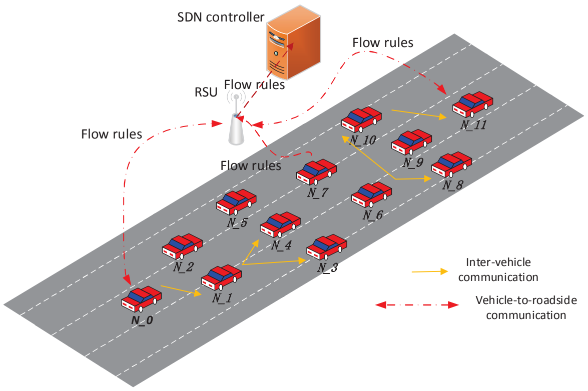

**Figure 1: An example of twelve-vehicle topology.**

***State***: The agent (SDN controller) interacts with the VANET environment including each vehicle's location and its forwarding ratio. We collect the location and forwarding ratio of each vehicle into two matrices: matrix of $L_N$ and matrix of $T_N$. The matrix $L_N$ can be given by

$$L_N = [L_0, L_1, L_2, ..., L_n]^\top \tag{1}$$

Meanwhile, we assume that each vehicle has $k$ states, and the matrix $T_N$ is shown as follow

$$T_N = \begin{bmatrix} T_0^0 & T_0^1 & T_0^2 & \cdots & T_0^k \\ T_1^0 & T_1^1 & T_1^2 & \cdots & T_1^k \\ T_2^0 & T_2^1 & T_2^2 & \cdots & T_2^k \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ T_n^0 & T_n^1 & T_n^2 & \cdots & T_n^k \end{bmatrix} \tag{2}$$

Specifically, because of the quality variance of communication channel, the forwarding ratio for each vehicle in the network has possibilities to change to another value. Let $p_n^{mm'} = p\{T_{n,t} = T_n^m | T_{n,t+1} = T_n^{m'}\}$, $m, m' = 1, ..., k$ represent the probability of vehicle $n \in N$ changing its state from $m$ (current moment is $t$) to $m'$ (at moment $t+1$), and $p_n^{mm'}$ satisfies the uniform distribution. So, the state transition probability of system state $p$ is set to be

$$p(s_t|s_{t+1}) = \prod_{n=0}^{N} p_n^{m_n m'_n} \quad m_n, m'_n \in 1, 2, ..., k \tag{3}$$

In summary, at time step $t$, the agent observes the environment and gets the state space $s_t = (L_N, T_N) \in S$ for the routing selection.

***Action***: The choice of the agent to decide the selection of any vehicle in the environment to forward packet to one of its neighbors is served as the set of action $a_t \in A$. In other words, the decision of the agent is defined as the selection of each vehicle's next hop neighbor directly connected state $s_t$. The value of action is between $[0, 1]$, which is related to the state space $S$.

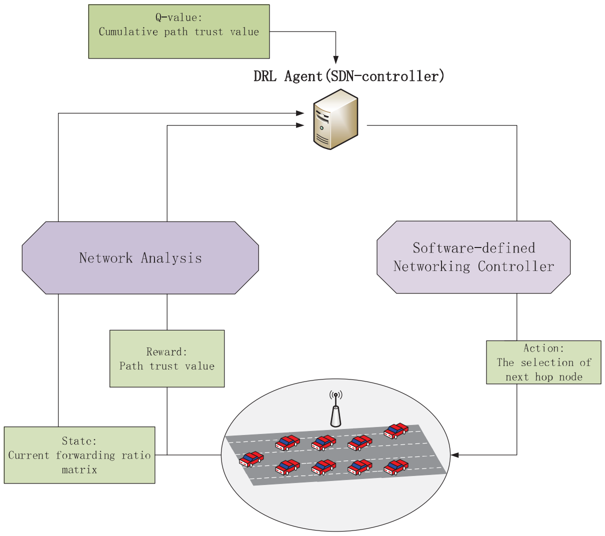
Q-value:
Cumulative path trust value
DRL Agent(SDN-controller)
Network Analysis
Software-defined Networking Controller
Reward:
Path trust value
Action:
The selection of next hop node
State:
Current forwarding ratio matrix

**Figure 2: The framework of proposed T-DDRL.**

***Reward***: In T-DDRL, the system reward is based on the trust information. In the trust model, each vehicle interacts with its neighbors to update their trustworthiness. Let vehicle $i, j \in N$, communicates with each other in a predetermine time interval $[t,\, t+\delta t]$. In other words, the trust value of vehicle $i, j$ is updated in a time interval $\delta t$ ($\delta t$<1) seconds. Meanwhile, a threshold $\lambda$ is introduced to determine the degree of trustworthiness of any vehicle in the network. If the trust value of a vehicle is greater than or equal to $\lambda$, this vehicle can be served as the trusted node. Otherwise, the vehicle is malicious node. Specifically, since the communication channel always keeps changing, the malicious nodes maybe convert to the trusted nodes according to the transition probability $p$.

Since the reward value is important for the communication of vehicles, we assume that the trust value of the source is set to 1, and it has no routing information towards to the destination. Moreover, the forwarding ratio is used to represent the trust value of each vehicle. The available range of vehicle trust value is [0,1].

In T-DDRL, the routing information can be divided into two groups: control routing information (RREQ, RREP and RRER) and data. Control packets determine the data transfer path in the path discovery procedure, and forwarding ratio of control packets is an important factor to determine vehicle trust value. According to [10], for vehicles $i, j$ at time step $t$, the computation of trustworthiness $V_{ij}(t)$ is shown below:

$$V_{ij}(t) = \omega_1 VT_{ij}^C(t) + \omega_2 VT_{ij}^D(t) \tag{4}$$

where $VT_{ij}^C(t)$ represents the trust value of control routing information and $VT_{ij}^D(t)$ represents the data packet direct trust. $V_{ij}(t)$ denotes the trust value of receiving vehicle $j$ for forwarding vehicle $i$. $\omega_1$ and $\omega_2$ are two weighted factors ($\omega_1, \omega_2 \geq 0$, and $\omega_1 + \omega_2 = 1$) that determine which forwarding ratio ($VT_{ij}^C(t)$ and $VT_{ij}^D(t)$) is more important in the vehicle trust calculating process. Particularly, in our proposed framework, we assume $\omega_1 = \omega_2$, which means that we consider path discovery and data forwarding process simultaneously.

According to [10], the forwarding ratio is calculated by the interaction between two neighbor vehicles. In our model, direct trust is the number of packets received correctly divided by the number of packets forwarded. The trust computation can be defined by the following formula:

$$VT_{ij}^{C}(t) = \frac{c_{ij}(t)}{t_{ij}(t)} \tag{5}$$

where $VT_{ij}^{C}(t)$ denotes the control routing information direct trust (forwarding ratio) for node $i$ towards its neighborhood $j$. $t_{ij}(t)$ denotes the total number of packets forwarded from node $i$ towards node $j$, and $c_{ij}(t)$ is the number of packets forwarded to next hop by node $j$ in the time period $t$. Similarly, the trust computation of $VT_{ij}^{D}(t)$ is same as the $T_{nn'}^{C}(t)$.

Based on the trustworthiness of each vehicle in the network, the immediate reward value $R_t \in R$ is decided by the quality of a link towards to the destination node. The trustworthiness of routing path from the source vehicle to the destination is being considered as immediate reward value. The path trust value needs to be computed according to the trust value of each vehicle along the path. Consequently, the final routing path from the source to the destination depends on the all vehicles' trust value on the route. Let a route $P$, consisting of $l$ nodes, representing as $N^{'} = \{0, 1, 2, ..., l\}$ and $N^{'} \in N$, where node $q$ denotes the $qth$ node in the route. So, the routing path trust value $R_P$ can be defined as follows:

$$R_P(t) = V_{0,1}(t)V_{1,2}(t)...V_{m-3,m-2}(t) = \prod_{q=0}^{l-3} NT_{q,q+1}(t) \tag{6}$$

In route $P$, the reward value $R_P(t)$ is served as the immediate reward from a vehicle $q$ to its neighbor $q+1$ at time instant $t$. The agent gets $R_P(t)$ in state $s_t$ when action $a_t$ is performed in time slot $t$. The goal of deep Q-network is to find an optimal policy to maximize the long term path trust value, and the cumulative reward can be written as:

$$R_P^{long} = \max E[\sum_{t=1}^{t=T} \gamma^t R_P(t)] \tag{7}$$

where $\gamma^t$ approaches to zero when $t$ is large enough. In our simulation, a threshold can be set for terminating the process.

***Path learning***: The agent aims to find an optimal routing from the source to the destination vehicle. Suppose the agent observes the state $s_t$ at the beginning time step $t$, then it decides which vehicle is the next hop to forward packets, and receives a sequence of rewards after time step $t$, $R_P(t), R_P(t+1), R_P(t+2), .....$ Q-value $Q(s_t, a_t)$ is a function for the state-action pair, and ending in the next state $s_{t+1}$. In T-DDRL, the path learning is that the Q-value is updated in its Q-table when selecting a next hop to forward packets.

The key insight behind our proposed framework is that we use dueling deep Q-network (DDQN) to evaluating the system Q-value. In T-DDRL, the SDN controller does not need to estimate the value of each action choice for state space $s_t$. In general, the Q-value function gives the expected total reward for corresponding state $s_t$ and action $a_t$ under policy $\pi$ (i.e., $\varepsilon - greedy$ policy) with discount factor $\gamma$. In our model, the state-action value $Q(s_t, a_t)$ is decomposed into two components as follows,

$$Q_\pi(s_t, a_t) = V_\pi(s_t) + A_\pi(s_t, a_t) \tag{8}$$

where $V_\pi(s_t)$ denotes the value of static states, and $A_\pi(s_t, a_t)$ represents the action advantage function, indicating the additional value of selecting an action under the state $s_t$.

## 3.2 Proposed Framework T-DDRL

In this section, we introduce the dueling deep reinforcement learning approach that extracts useful features from the routing information and finds the optimal policy (the most trusted routing path) $\pi^*$. The experience replay and the target Q-network are used to improve the stability of proposed scheme.

In T-DDRL, the end-to-end goals are achieved by the path learning and trust computation process. As shown in Fig. 2, the framework mainly includes three important components: SDN controller, SDN-enabled vehicles, and SDN-enabled RSU.

***Framework description***: The proposed T-DDRL is initialized when the source vehicle needs to build the communication with the destination vehicle. Firstly, the source launches path discovery process to establish data transfer path. We assume that in the path discovery process, each vehicle's trust value is unchanged. In this phase, each node initializes path learning and trust computation to select the best policy for the data transfer path. The path discovery procedure aims to establish initial trust value for each vehicle and to find a best routing path to forward data. Since the transition probability for each vehicle's state, any vehicle's trust value has possibilities to change according to $p$ in data forwarding process, so the established routing path may change to untrusted path. Consequently, the agent needs to learn a new policy according to the different state and position value of each vehicle in the network environment.

***DQN architecture***: We build our DDQN following the deep neural network as 7, 8, and 9 hidden layers, where the network input is the state $s_t = (L_N, T_N)$ and the output is a vector of estimated Q-value $Q(s_t, a_t; \theta, \alpha, \beta)$. The DDQN architecture for T-DDRL is given in Fig. 3, where we make one stream of hidden layer 9 output a scalar $V(s_t; \theta, \beta)$, and the other stream output a vector $A(s_t, a_t; \theta, \alpha)$. Therefore, the final Q-value in T-DDRL is defined as follows

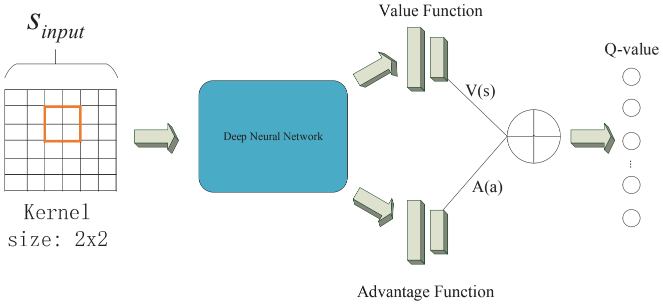

**Figure 3: Deep neural network for T-DDRL training.**

$$Q_\pi(s_t,a_t;\theta,\alpha,\beta)=V_\pi(s_t;\theta,\beta)+\Bigg(A_\pi(s_t,a_t;\theta,\alpha)+\frac{1}{|A|}\sum_{a_{t+1}}A_\pi(s_t,a_{t+1};\theta,\alpha)\Bigg) \tag{9}$$

where $\theta$ denotes the parameters of the hidden layers, while $\alpha$ and $\beta$ are the parameters of the two streams of final output layers.

In T-DDRL, we aim to optimize the loss function $L(\omega)$ using *mean square loss* (MSE)

$$MSE_{L(\omega)}=\frac{1}{t}\sum_{i=0}^{t}\Bigg(R_P(t)+\gamma\max_{a_{i+1}}Q(s_{i+1},a_{i+1};\theta^-,\alpha^-,\beta^-)-Q(s_i,a_i,\theta,\alpha,\beta)\Bigg)^2 \tag{10}$$

***DQN training***: In each time step $t$, the agent observes state-action pair $E_t=\{s_t,a_t,R_P(t),s_{t+1}\}$ from the network environment, and stores the $E_t$ into the replay memory $D=\{E_1,E_2,E_3,...,E_t\}$. Since memory $D$ is a finite capacity, the oldest data will be discarded when the memory is full. Moreover, $D$ always keeps updating its data when a new state-action pair is coming. In order to get the optimal Q-value $Q^*(s_t,a_t;\theta,\alpha,\beta)$, the agent needs to train data including: input data matrix $s_t=(L_N,T_N)$, and corresponding targets value $y=Q^*(s_t,a_t;\theta,\alpha,\beta)$. The input data can get from the replay memory $D$. Because of the unknown target value, we can use estimated value $y=R_P(t)+\gamma\max_{a_{t+1}}Q(s_{t+1},a_{t+1};\theta^-,\alpha^-,\beta^-)$ as our target Q-value. The agent randomly samples from the experience replay memory to form the input data to train the network. After the agent is trained, it will reach a good estimated optimal Q-value and find an optimal routing selection policy. We will discuss the simulation results in the following section.

## 4 SIMULATION RESULTS AND DISCUSSIONS

In this section, we describe our simulation setup, configurations, and simulation results. We simulate our work in TensorFlow (version 1.6.0)[1] and OPNET. In our simulation, there are two types of nodes: i) normal nodes, which forward the data packets correctly; ii) malicious nodes, which randomly drop the received data packets. Specifically, we set the number of malicious nodes much smaller compared with the number of normal nodes. We apply deep neural networks with different hidden layers as deep Q-network to compare the training efficiency of path learning. Moreover, we verify our proposed T-DDRL by simulation in terms of average network throughput and end-to-end delay.

**Table 1: Simulation Setup**

| Simulation Parameter | Assigned value |
|---|---|
| Topology | Random and grid |
| Packet size | 1024 bytes |
| IEEE 802.11 MAC | 802.11a |
| Topology covered area | $5\times 5km^2$. |
| Data Rates | 1Mbps, 2Mbps, 5.5Mbps, 11Mbps |
| Mobility | Static (none) |
| Number of nodes | 8, 12, 20, 24, 28 and 32 |
| Network | OpenFlow |
| Simulation time | 15 mins |

### 4.1 Simulation Setup

The simulation has been implemented in the processor of Intel(R) Core(TM) i7-6600 CPU with 16GB memory. The software environment that we use is TensorFlow 1.6.0 with python 3.6 and OPNET 14.5 on Windows 10 64-bit operating system.

In our simulation, the proposed scheme with deep neural networks is compared with two existing schemes: i) In the existing scheme with trust based software-defined networking for VANETs. The main problem of the existing scheme is that it cannot respond a situation that any trusted vehicle in the network changes to the malicious nodes, ii) The original AODV protocol.

Meanwhile, SDN-enabled vehicles, one RSU, and one centralized SDN controller are randomly deployed within the covered area. We assume that the vehicles' state can be good ($V_{ij}(t)\geq\lambda$) or bad ($V_{ij}(t)<\lambda$). Good vehicles are trusted nodes that aim to establish secure routing path from the source to the destination. Conversely, bad vehicles are served as the malicious nodes that will degrade the network performance. Meanwhile, we set the transition probability of the source and the destination vehicle to 1. An example of the transition probability matrix for an intermediate vehicle in the network environment can be set as follows:

$$p=\begin{bmatrix}0.2&0.1&0.3&0.1&0.3\\0.1&0.2&0.3&0.1&0.3\\0.2&0.1&0.1&0.3&0.3\\0.3&0.3&0.2&0.1&0.1\\0.1&0.3&0.2&0.1&0.3\end{bmatrix} \tag{11}$$

The deep Q-network used in this paper is normal neural network with 7, 8, and 9 hidden layers. In our simulation,

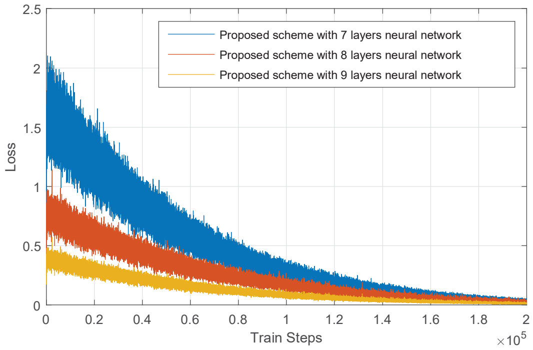

**Figure 4: Convergence performance versus different architectures of DQN.**

the architecture of evaluation network is same as the target Q-network. However, only the evaluation network is trained depending on the gradient descent method, and we replace the target Q-network by trained Q-value every 5 steps (after 200 steps).

The values of the rest parameters are summarized in Table I.

## 4.2 Simulation Results

Fig. 4 shows the convergence performance comparison of different scenarios in the proposed schemes using deep reinforcement learning. In our simulation, loss function $L(\omega)$ reflects the degree of Q-value fitting. As shown in Fig. 4, we can observe that the value of $L(\omega)$ in deep neural network (DNN) with 7 hidden layers is higher than the schemes with 8 and 9 hidden layers. Therefore, as the hidden layers increase, the degree of data fitting becomes better. We can see that when the number of hidden layers of the deep neural network is 8 and 9, they all converge within the predetermined episodes. The best performance is our proposed method using 9 hidden layers since the rate of convergence is the fastest compared with other two methods. Meanwhile, the value of loss function using 9 hidden layers is lower than the other three methods, and it reflects the data fitting degree has the best performance.

Fig. 5 shows the convergence performance of the proposed scheme using dueling deep neural network (nine hidden layers) with different learning rates. Learning rate is an important factor that determines the updating speed of weighted factor in deep Q-network. If the learning rate is not appropriate for current deep neural network, it will deeply interfere with the Q-network convergence. From Fig. 5, we can conclude that a higher learning rate will increase the degree of oscillation of loss function and badly influence the algorithm convergence. As shown in Fig. 5, when the learning rate equals to $9 \times 10^{-7}$, T-DDRL does not converge in the predetermined episodes, and has highly oscillation. In the rest of simulation, we choose the final learning rate of 0.00000001.

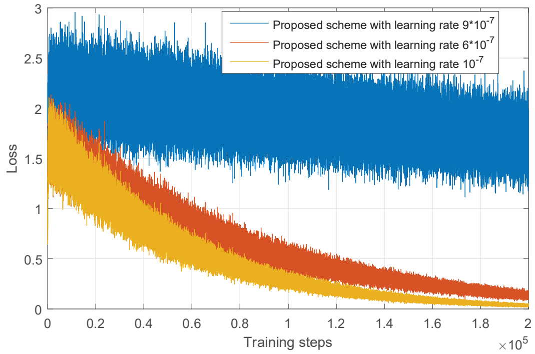

**Figure 5: Convergence performance versus different learning rates of DQN.**

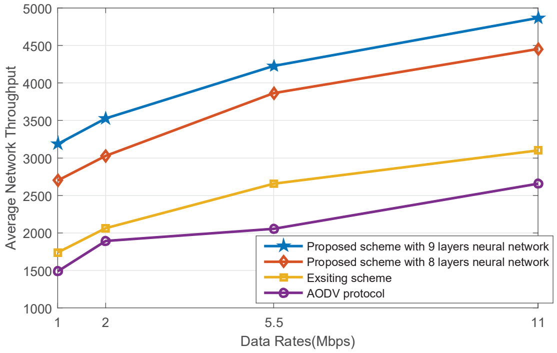

**Figure 6: Throughput comparison versus different data rates.**

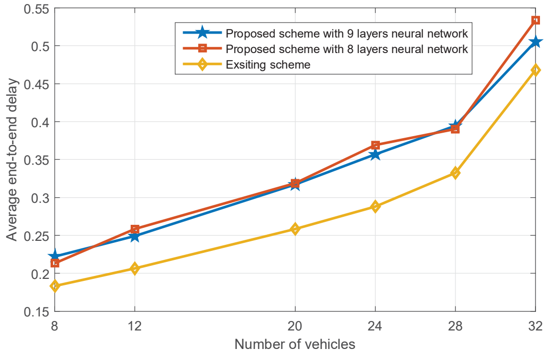

**Figure 7: Average end-to-end delay compasiton versus different number of vehicles.**

Fig. 6 shows the comparison of average network throughput in different data rates. The network throughput is an important factor that evaluates the network performance. Since the malicious vehicles will deeply interfere the network throughput[18], the network throughput of four schemes all decreases as the data rate grows. However, the proposed

T-DDRL method is still better than the other two schemes. This is because our proposed schemes aim to maximize the long-term reward $R_P(t)$. The path learning process keeps learning a most trusted routing path. The final selected vehicles in the path can forward more routing information to destination.

Fig. 7 shows the comparison of average end-to-end delay of four schemes within the different number of vehicles. From the figure, we can conclude that the proposed scheme has a slightly higher average end-to-end delay than the existing scheme as the number of vehicles grows. This is because the selected routing path in T-DDRL is not intended to find a minimum hop route, so the trusted based route is usually a longer routing path from a source node to a destination node. Therefore, a trivial delay is introduced by the proposed scheme compared with the existing scheme. However, higher security and intelligence are achieved in the proposed scheme.

## 5 CONCLUSION

In this paper, we presented a dueling deep reinforcement learning approach used in a VANET environment. We designed a software-defined trust based dueling deep reinforcement learning(T-DDRL) approach. In the trust computation phase of T-DDRL, we introduced a trust model to decide the immediate path trust for the long-term reward (Q-value) in the path learning. In the path learning process, we used dueling deep Q-learning algorithm to determine the best routing policy. The centralized SDN control platform acts as an agent to interact with the environment. The simulation results show the effectiveness of our proposed method. In the future, we hope to use multi-agent deep reinforcement learning approach to enhance the performance.

## ACKNOWLEDGMENT

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) DIVA Network.

## REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[2] Khadige Abboud and Weihua Zhuang. 2014. Impact of node mobility on single-hop cluster overlap in vehicular ad hoc networks. In *Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile systems*. 65–72.

[3] PS Abi, M Devi, and V Rhymend Uthariaraj. 2018. Collaborative trust-based security and power control techniques for VANET. *International Journal of Mobile Network Design and Innovation* 8, 2 (2018), 65–72.

[4] Sami S Albouq and Erik M Fredericks. 2017. Detection and Avoidance of Wormhole Attacks in Connected Vehicles. In *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*. ACM, 107–116.

[5] Gautam M Borkar and AR Mahajan. 2017. A secure and trust based on-demand multipath routing scheme for self-organized mobile ad-hoc networks. *Wireless Networks* 23, 8 (2017), 2455–2472.

[6] Saloua Chettibi and Salim Chikhi. 2016. Dynamic fuzzy logic and reinforcement learning for adaptive energy efficient routing in mobile ad-hoc networks. *Applied Soft Computing* 38 (2016), 321–328.

[7] Khalid Abdel Hafeez, Lian Zhao, Bobby Ma, and Jon W Mark. 2013. Performance analysis and enhancement of the DSRC for VANET's safety applications. *IEEE Transactions on Vehicular Technology* 62, 7 (2013), 3069–3083.

[8] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. 2015. Software-defined networking: A comprehensive survey. *Proc. IEEE* 103, 1 (2015), 14–76.

[9] Mushu Li, Lian Zhao, and Hongbin Liang. 2017. An SMDP-based prioritized channel allocation scheme in cognitive enabled vehicular ad hoc networks. *IEEE Trans. Veh. Technol* 66, 9 (2017), 7925–7933.

[10] Xin Li, Zhiping Jia, Peng Zhang, Ruihua Zhang, and Haiyang Wang. 2010. Trust-based on-demand multipath routing in mobile ad hoc networks. *IET Information Security* 4, 4 (2010), 212–232.

[11] Arun Malik and Babita Pandey. 2018. Security Analysis of Discrete Event Based Threat Driven Authentication Approach in VANET Using Petri Nets. *IJ Network Security* 20, 4 (2018), 601–608.

[12] Daniel Markert, Philip Parsch, and Alejandro Masrur. 2017. Using Probabilistic Estimates to Guarantee Reliability in Crossroad VANETs. In *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*. ACM, 135–142.

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529.

[14] Prateek Kumar Singh, Koushik Kar, James H Nguyen, and Daniel Ku. 2017. Mass Configuration with Confirmation in Tactical Networks. In *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications*. ACM, 99–106.

[15] Shrikant Tangade, Sunilkumar S Manvi, and Pascal Lorenz. 2018. Decentralized and Scalable Privacy-Preserving Authentication Scheme in VANETs. *IEEE Transactions on Vehicular Technology* (2018).

[16] Yanwei Wang, F Richard Yu, Minyi Huang, Azzedine Boukerche, and Tao Chen. 2013. Securing vehicular ad hoc networks with mean field game theory. In *Proceedings of the third ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*. 55–60.

[17] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. 2015. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581* (2015).

[18] Zhexiong Wei, Helen Tang, F Richard Yu, Maoyu Wang, and Peter Mason. 2014. Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning. *IEEE Transactions on Vehicular Technology* 63, 9 (2014), 4647–4658.

[19] Zhexiong Wei, F Richard Yu, Helen Tang, Chengchao Liang, and Qiao Yan. 2016. Security Schemes in Vehicular Ad hoc Networks with Cognitive Radios. *arXiv preprint arXiv:1611.06905* (2016).

[20] Wenfeng Xia, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, and Haiyong Xie. 2015. A survey on software-defined networking. *IEEE Communications Surveys & Tutorials* 17, 1 (2015), 27–51.

[21] Qiao Yan, F Richard Yu, Qingxiang Gong, and Jianqiang Li. 2016. Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Communications Surveys & Tutorials* 18, 1 (2016), 602–622.

[22] F Richard Yu. 2016. Connected Vehicles for Intelligent Transportation Systems [Guest editorial]. *IEEE Transactions on Vehicular Technology* 65, 6 (2016), 3843–3844.

**DOCUMENT CONTROL DATA**

*Security markings for the title, authors, abstract and keywords must be entered when the document is sensitive

1. ORIGINATOR (Name and address of the organization preparing the document. A DRDC Centre sponsoring a contractor's report, or tasking agency, is entered in Section 8.)

Association for Computing Machinery
1601 Broadway, 10th Floor
New York, NY 10019-7434
www.acm.org/

2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.)

CAN UNCLASSIFIED

2b. CONTROLLED GOODS

NON-CONTROLLED GOODS
DMC A

3. TITLE (The document title and sub-title as indicated on the title page.)

A Deep Reinforcement Learning-based Trust Management Scheme for Software-defined Vehicular Networks

4. AUTHORS (Last name, followed by initials – ranks, titles, etc., not to be used)

Zhang, D.; Yu, F.R.; Yang, R.;  Tang, H.

5. DATE OF PUBLICATION (Month and year of publication of document.)

October 2018

6a. NO. OF PAGES (Total pages, including Annexes, excluding DCD, covering and verso pages.)

7

6b. NO. OF REFS (Total references cited.)

22

7. DOCUMENT CATEGORY (e.g., Scientific Report, Contract Report, Scientific Letter.)

External Literature (N)

8. SPONSORING CENTRE (The name and address of the department project office or laboratory sponsoring the research and development.)

DRDC – Ottawa Research Centre
Defence Research and Development Canada
3701 Carling Avenue
Ottawa, Ontario K1A 0Z4
Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

Legacy --> Project --> 7 - Centre for Security Sciences

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. DRDC PUBLICATION NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC-RDDC-2022-N286

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11a. FUTURE DISTRIBUTION WITHIN CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

Public release

11b. FUTURE DISTRIBUTION OUTSIDE CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

12. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Use semi-colon as a delimiter.)

Networks; Network Protocols; Network Layer Protocols; Routing Protocols

13. ABSTRACT/RÉSUMÉ (When available in the document, the French version of the abstract must be included here.)

Vehicular ad hoc networks (VANETs) have become a promising technology in intelligent transportation systems (ITS) with rising interest of expedient, safe, and high-efficient transportation.

VANETs are vulnerable to malicious nodes and result in performance degradation because of dynamicity and infrastructure-less. In this paper, we propose a trust based dueling deep reinforcement learning approach (T-DDRL) for communication of connected vehicles, we deploy a dueling network architecture into a logically centralized controller of software-defined networking (SDN). Specifically, the SDN controller is used as an agent to learn the most trusted routing path by deep neural network (DNN) in VANETs, where the trust model is designed to evaluate neighbors’ behaviour of forwarding routing information. Simulation results are presented to show the effectiveness of the proposed T-DDRL framework.