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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

A Reinforcement Learning-based Trust Model for Cluster Size Adjustment Scheme in Distributed Cognitive Radio Networks

Mee Hong Ling, *Member, IEEE,* Kok-Lim Alvin Yau, *Member, IEEE,* Junaid Qadir, *Senior Member, IEEE,* and Qiang Ni, *Senior Member, IEEE*

***Abstract*—Cognitive radio enables secondary users (SUs) to explore and exploit the underutilized licensed channels (or white spaces) owned by the primary users. To improve the network scalability, the SUs are organized into clusters. This article proposes a novel artificial intelligence based trust model approach that uses reinforcement learning (RL) to improve traditional budget-based cluster size adjustment schemes. The RL-based trust model enables the clusterhead to observe and learn about the behaviors of its SU member nodes, and revoke the membership of malicious SUs in order to ameliorate the effects of *intelligent* and *collaborative* attacks, while adjusting the cluster size dynamically according to the availability of white spaces. The malicious SUs launch attacks on clusterheads causing the cluster size to become inappropriately sized while learning to remain undetected. In any attack and defense scenario, both the attackers and the clusterhead adopt RL approaches. Simulation results have shown that the single-agent RL (SARL) attackers have caused the cluster size to reduce significantly; while the SARL clusterhead has slightly helped increase its cluster size, and this motivates a rule-based approach to efficiently counterattack. Multi-agent RL attacks have shown to be less effective in an operating environment that is dynamic.**

***Index Terms*—Cognitive radio ad hoc networks, clustering methods, trust management, artificial intelligence, security.**

I. Introduction

COGNITIVE radio (CR) networks consist of unlicensed users (or secondary users, SUs) that explore and use underutilized licensed channels (or white spaces) owned by licensed users (or primary users, PUs) in an opportunistic manner [1]. In a CR network (CRN), where the availability of white spaces is unpredictable and the amount of resources are limited, the need to utilize them in an efficient manner is desirable, and one of the ways to improve such efficiency is through clustering. Clustering organizes SUs into logical groups; a SU acts as a leader (or *clusterhead*) that serves as a local point of process for essential CR operations, such as channel sensing and routing, or a *member node* that associates itself with the clusterhead. Leveraging on clustering reduces the signaling overhead in the network, improves routing efficiency by having minimum nodes in the backbone network, reduces update on routing information cost, and eliminates the need to update the PUs' activities network wide [2], [3]. With these significant overhead reductions, the white spaces can be used to enhance network scalability [4], which implies efficient resource utilization [5], [6]. A network is considered scalable when the number of member nodes in a cluster increases or the number of clusters in the network decreases.

This work was supported by the Malaysian Ministry of Education under Fundamental Research Grant Scheme FRGS/1/2014/ICT03/SYUC/02/2 and Sunway-Lancaster Grant Scheme SGSSL-FST-CSNS-0114-05 and PVM1024.

M. H. Ling and K.-L. A. Yau are with the Department of Computing and Information Systems, Faculty of Science and Technology, Sunway University, Selangor, Malaysia. (e-mail: mhling@sunway.edu.my, koklimy@sunway.edu.my)

J. Qadir is with the Information Technology University (ITU), Lahore, Pakistan. (e-mail: junaid.qadir@itu.edu.pk)

Q. Ni is with the School of Computing and Communications, Lancaster University, Lancaster, United Kingdom. (e-mail: q.ni@lancaster.ac.uk)

A. Attacker and Clusterhead

In our article, there are two main entities of interest: the potential attackers (SUs) and the clusterhead. We consider all SUs as potential attackers and there is no specific criteria or a threshold to differentiate an attacker from a legitimate (or regular) SU. The following sub-sections provide a general overview of their roles in CRNs.

1) Attacker's Role: In our attack scenario, the attackers have their own attack capability, which is measured in terms of attack probability $0.1 \leq P_t^i \leq 0.9$ and attack intensity $0.1 \leq I_t^i \leq 1$. Note that $P_t^i = 1$ is not considered in order to provide some irregularity or randomness in the attacks. The main objective of the attackers is to waste the granted/acquired resources, thereby compromising the network scalability. This kind of attack has its advantage over the traditional attacks, such as random and deterministic attacks [7], as it requires the malicious SUs to learn from the operating environment and take an appropriate action based on their observations.

2) Clusterhead's Role: In our counterattack scenario, the clusterhead adopts a trust model to establish trust amongst its member nodes. The trust model is incorporated into the cluster size adjustment scheme, which is embedded in the clusterhead. The main objective of the clusterhead is to learn the nodes' behavior and select the legitimate nodes as member nodes in order to maximize its resource utilization.

In our article, the attacker's model is formulated using reinforcement learning (RL) [8]. RL is an unsupervised artificial intelligence approach that enables a decision maker (or an agent) to observe its operating environment represented by *state*, and to measure the positive or negative effects of its *action* at that state represented by *delayed reward*; this enables the agent to learn, and subsequently select a proper action to be carried out. Using RL, a SU can gain experience and learn

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

about the best possible action for a particular state in order to maximize accumulated rewards as time goes by. Hence, an attacker that launches an intelligent attack has the capacity to learn the effects of its attacks on the clusterhead (the reduction of the cluster size) while avoid being detected. For instance, when an attacker does not receive the requested amount of resources (i.e., clusterhead either grants partial amount or no resources at all), it observes the clusterhead's perceived trust (i.e., state) on its earlier action (i.e., attack probability P_t^i and attack intensity I_t^i). In our work, we investigate two main types of intelligent attacks, namely an independent attack known as single-agent reinforcement learning (SARL) attack, and a collaborative attack known as multi-agent reinforcement learning (MARL) attack, which involves message (or payoff) exchange amongst a group of attackers.

The trust model for the cluster size adjustment scheme is also formulated and solved using RL. The trust model is defined in the form of Q-values to keep track of member nodes' malicious behavior, which may change as time goes by. Using the Q-values, the clusterhead selects the legitimate nodes represented by *action* and include them in the cluster. For instance, when the clusterhead observes that its selected member node (i.e., action) utilizes the granted resources, its trust value increases. The effect of the clusterhead's action (i.e., delayed reward) is an increase in the cluster size.

B. Related Work

While the existing work or approaches have provided some scalability in clustering, investigation into the effects of intelligent attacks, namely RL attacks on clusterhead, and RL-based trust countermeasures is lacking. In [9], [10], given a budget value (i.e., the white spaces amount), a cluster size adjustment scheme allows a clusterhead to distribute the budget value to its neighboring nodes in the form of tokens. Since a neighboring node must receive a token to join a cluster and become its member node, the clusterhead is able to manage its cluster size to its intended upper bound cluster size, which implies cluster scalability.

In [11], attackers launch PU emulation (PUE) attacks on cooperative spectrum sensing in CRNs so that a clusterhead makes wrong cluster-level decisions about the presence of PUs' activities, and so the clusterhead adopts a reputation-based hierarchically cooperative spectrum sensing scheme to improve detection performance and to maintain a comparatively low communication overhead. In [12], attackers launch jamming attacks on the control channel of multi-channel ad hoc networks, and so the nodes in the clusters adopt a randomized and distributed scheme based on frequency hopping to establish a new control channel. In [13], a resource allocation scheme is used to provide energy efficiency by selecting legitimate SUs in CRNs in the presence of PUE attacks.

In [14]–[19], trust models have been applied to channel sensing and channel access in CRNs to detect malicious nodes in order to improve sensing outcomes with reduce missed detection and false alarm rates. In [20], a reputation scheme has been applied to a clusterhead to detect its malicious member nodes in order to provide higher security to the non-manipulability and the agreement property of clusterhead election results, and consequently it maintains the network performance with the increase of member nodes. In [21], a comprehensive survey has been done on various trust models in CRNs to ameliorate the effects of malicious SUs launching collaborative attacks.

C. Motivation and Significance

In our article, cluster size adjustment is investigated in the context of CR to provide network scalability. Larger cluster size (or smaller number of clusters in a CRN) has been shown to improve network scalability by reducing routing overheads [22], while smaller cluster size has been shown to increase the number of common channels in a cluster [23]. Hence, both too small or too large a cluster are unfavorable — a small cluster indicates resource underutilization while a big cluster indicates resource deprivation of the member nodes. By using the budget value, the cluster size can be prevented from getting too small or large. Therefore, to improve network scalability, a clusterhead ensures that the tokens distributed to its neighboring nodes are as close to the available budget value as possible. The tokens serve as a representation of the rightful or legitimate users to resource utilization. A token may represent a unit or several unit of resources. Most importantly, the tokens are given to the *legitimate* neighboring nodes so that the white spaces are well utilized, otherwise the tokens will be wasted. We consider a member node as *legitimate* when it has been given transmission opportunities and is entrusted with additional resources by the clusterhead, and a member node as *malicious* when it is not given any resources at that time of instance.

Due to the dynamicity of channel availability in CRNs, the member nodes must obey the clusterhead the clusterhead in order to utilize the white spaces efficiently. The requirement to fully utilize the granted tokens by its member nodes, has brought about security challenges in clustering. This is because the member nodes may become malicious over time and may launch attacks in an *intelligent* manner. As the clusterhead plays a critical role in clustering, it is vital for the clusterhead to counter intelligent attacks in order to ensure that the white spaces are appropriately utilized. While the attackers launch intelligent attacks to create a more detrimental effect to the clusterhead, the clusterhead also counters the attack by leveraging on RL-based trust model for cluster size adjustment to provide cluster scalability. In [24], an in-depth study was done on cluster size adjustment in the absence of attackers. While some research has been done on cluster size adjustment to improve network scalability, to the best of our knowledge, the study on the effects of *intelligent* attacks, which are detrimental to network scalability and to the clusterhead has not been carried out; and this is the primary motivation for our article.

The significance of our work is summarized as follows:

- *Intelligent attacks with varying probability and intensity of attack* are considered to avoid being detected by the clusterhead.

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

- *Intelligent countermeasure*, which is a combination of rules and learning, is adopted to learn the attackers' actions and so remove them from the cluster.

D. Our Contributions

The novel contributions of our article are as follows:

(a) To propose a SARL-based trust model for cluster size adjustment scheme to be applied to counter intelligent and collaborative attacks. The scheme helps the cluster to grow to its appropriate size (i.e., the size given by the budget value) in order to maximize the utilization of white spaces, hence achieving network scalability. The SARL-based trust model is embedded in a clusterhead to keep track of the trust value of each member node, which is then used for token distribution: member nodes with the highest trust value will be granted the full amount of tokens, while member node with the lowest trust value may not be granted the full amount of tokens or any at all, if the amount of budget is limited (i.e., the amount of white spaces is insufficient). To the best of our knowledge, this kind of trust model has not been applied to clustering in CRNs, particularly for cluster size adjustment scheme to detect malicious SUs in order to increase the utilization of white spaces leading to increased cluster scalability.

(b) To propose conditions for legitimate clusterhead to improve the performance of the SARL-based trust model where RL algorithms with different capabilities are applied to malicious SUs. These conditions are categorized into *three* cases for performance analysis as follows:
- Case I: Both legitimate clusterhead and malicious SUs adopt the SARL approach.
- Case II: The legitimate clusterhead uses a RL approach with greater capability compared to malicious SUs particularly, i.e., the legitimate clusterhead uses rule-based SARL, which only allows SUs to join the cluster when they have met or exceeded certain Q-value, and malicious SUs use SARL approach.
- Case III: Malicious SUs use a RL algorithm with greater capability compared to legitimate clusterhead particularly, i.e., malicious SUs use MARL, and legitimate clusterhead uses SARL.

E. Organization of this Article

Section II presents CRN, PU and attack models, white space computation, and cluster size adjustment. Section III presents problem formulation for both the attacker and the clusterhead. Section IV presents the generic SARL and MARL algorithms. Section V presents the RL-based attack model and trust model for cluster size adjustment. Section VI discusses the simulation overview, analyses the random and RL-based attacks, and presents performance evaluation, results and discussion. Section VII presents the main conclusions of our work and provides a list of future work for consideration.

TABLE I: General notations

Notation	Description
$t \in T$	A decision epoch (also called a time window) with $T = \{1, 2, \dots\}$.
$t_s \in T_s$	A time slot with $T_s = \{1, 2, \dots, \lvert T_s \rvert\}$.
$m \in M$	A PU with $M = \{1, 2, \dots, \lvert M \rvert\}$. PU m occupies licensed channel m.
$i \in \mathcal{I}$	A SU with $\mathcal{I} = \{1, 2, \dots, \lvert \mathcal{I} \rvert\}$.
$j \in J^i$	A neighboring node set of SU i.
$F^i \subseteq M$	A channel set of SU i.
r_m	The transmission range for PU m.
w^i_{m,t_s}	A unit of white space for channel m available to SU i at a time slot t_s.
w^i_t	Total amount of white spaces available to SU i at decision epoch t.
Λ_{m,t_s}	Availability of channel m at decision epoch t. $\Lambda_{m,t_s} = 1$ means PU is idle or OFF at time slot t_s, and $\Lambda_{m,t_s} = 0$ means PU is busy or ON at time slot t_s.
Λ^i_{m,t_s}	Equivalent of Λ_{m,t_s} for SU i.
$\mathcal{P}_{m,t_s}$	Transition probability matrix of PU activity in channel m at time slot t_s.
$p^{d\to b}_{m,t_s}$	$p^{d\to b}_{m,t_s} = \text{Prob}\{\Lambda_{m,t_s} = 0 \lvert \Lambda_{m,t_s-1} = 1\}$ is the probability of channel m switches from state OFF at time slot $t_s - 1$ to ON at time slot t.
$p^{b\to d}_{m,t_s}$	$p^{b\to d}_{m,t_s} = \text{Prob}\{\Lambda_{m,t_s} = 1 \lvert \Lambda_{m,t_s-1} = 0\}$ is the probability of channel m switches from state ON at time slot $t_s - 1$ to OFF at time slot t.
D^i_t	Performance indicator of SU i at decision epoch t.
P^i_t	Probability of attack launched by SU i at decision epoch t.
I^i_t	Intensity of attack launched by SU i at decision epoch t.
$\mathbb{P}_c$	Probability of attack in cluster c.
$\mathbb{I}_c$	Intensity of attack in cluster c.

TABLE II: Clustering notations

Notation	Description
$c \in C$	A cluster with $C = \{1, 2, \dots\}$.
$\mathcal{T}$	Units of available budget in the form of tokens.
$\beta^{avail}_{c,t}$	The available budget for cluster c at decision epoch t.
$\beta^{request}_{i,t}$	Requested tokens from member node i at decision epoch t.
$\beta^{granted}_{i,t}$	Granted tokens to member node i at decision epoch t.
$\beta^{unused}_{c,t}$	The unused budget for cluster c at decision epoch t. A budget is considered unused when the member node does not utilize the budget.
$\beta^{used}_{c,t}$	The used budget for cluster c at decision epoch t.
CH_c	The clusterhead of cluster c.
$Z_{c,t}$	Cluster size of cluster c at decision epoch t.
$\mathbb{R}_{c,t}$	Cluster size ratio of cluster c at decision epoch t.
$MN^{i,h}$	A member node i at h hops from its clusterhead.
h	The number of hops away from a clusterhead with $h = \{1, 2, \dots\}$.

II. System Model

In our system model, we assume that: a) the clusterhead is a legitimate entity and it aims to grant tokens to the legitimate SUs at each decision epoch t and in return the legitimate SUs are expected to fully utilize the given tokens so as to maximize the network resource utilization, and b) malicious SUs launch

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

TABLE III: RL notations

Notation	Description
$s_t^i \in S$	State of SU i at decision epoch t.
$a_t^i \in A$	Action of SU i at decision epoch t.
$r_{t+1}^i(s_{t+1}^i)$	Delayed reward of SU i under state s_{t+1}^i at decision epoch $t+1$.
$Q_t^i(s_t^i, a_t^i)$	Q-value of SU i under state s_t^i and action a_t^i at decision epoch t.
α	Learning rate with $0 < \alpha < 1$.
γ	Discount factor with $0 < \gamma < 1$.
$\eta^{i,j}$	Weight of information exchanged between SUs i and j.

attacks without interfering the PUs. This section is organized as follows. Section II-A presents the cognitive radio network and primary user models. Section II-B presents the white space computation. Section II-C presents the cluster size adjustment scheme. Section II-D presents the attack model.

A. Cognitive Radio Network and Primary User Models

We consider a time-slotted system with a decision epoch being $t \in T = \{1, 2, \dots\}$. Each decision epoch t consists of time slot $t_s \in T_s = \{1, 2, \dots, |T_s|\}$ with $|T_s|$ being the number of time slots in a decision epoch. Generally speaking, the availability of white spaces depends on the amount of time slots t_s in each decision epoch t. The greater the amount of time slots t_s, the higher the amount of available transmission resources. A node that needs more than one decision epoch t can be seen as requiring more transmission resources. The decision epoch repeats, and so a node can transmit more in different decision epochs. In our work, we have made decision epoch t configurable in order to cater for various transmission needs, and this does not affect the system model and our simulation work. Nodes $\mathcal{I} = \{1, 2, \dots, |\mathcal{I}|\}$ are independently and randomly distributed. Upon joining a cluster, a node i becomes a h-hop member node $MN^{i,h} \in \mathcal{I} = \{1, 2, \dots, |\mathcal{I}|\}$ in a cluster c and it transmits data packets to its clusterhead CH_c in the allotted white spaces, which is a transmission opportunity that allows the node to transmit.

Each PU $m \in M = \{1, 2, \dots, |M|\}$ occupies a licensed channel m, and it has a particular transmission range r_m. The activity of PU m is represented by a two-state ON-OFF (or busy-idle) Markov chain with transition probability matrix $\mathcal{P}_{m,t_s} = \begin{pmatrix} 1-p_{m,t_s}^{d\to b} & p_{m,t_s}^{d\to b} \\ p_{m,t_s}^{b\to d} & 1-p_{m,t_s}^{b\to d} \end{pmatrix}$ where $p_{m,t_s}^{d\to b} = \text{Prob}\{\Lambda_{m,t_s} = 0 | \Lambda_{m,t_s-1} = 1\}$ indicates the probability of PU m switches from OFF (idle) state at time slot $t_s - 1$ to ON (busy) state at time slot t_s, and $p_{m,t_s}^{b\to d} = \text{Prob}\{\Lambda_{m,t_s} = 1 | \Lambda_{m,t_s-1} = 0\}$ indicates the probability of PU m switches from ON (busy) state at time slot $t_s - 1$ to OFF (idle) state at time slot t_s [25].

Each SU $i \in \mathcal{I} = \{1, 2, \dots, |\mathcal{I}|\}$ has a different set of available channels $F^i \subseteq M$. The availability of channel m at SU i can be affected if it is physically located within the transmission range r_m of PU m. Let $\Lambda_{m,t_s}^i = 1$ indicates that channel m of SU i is absent of PUs' activity or idle (or the OFF state) at time slot t_s. Given the transition probability matrix of PU m, SU i, which is physically located within a PU's transmission range r_m in the absence of PU (i.e., $\Lambda_{m,t_s}^i = 1$), can opportunistically utilize white space w_{m,t_s}^i from channel m at time slot t_s. When SU i is physically located out of a PU's transmission range r_m, channel m is available at all times to SU i. We define the availability of channel m for SU i at timeslot t_s as follows:

$$w_{m,t_s}^i = \begin{cases} 1, & \text{if } \Lambda_{m,t_s}^i = 1 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

B. White Space Computation - the Budget Value

A clusterhead CH_c manages cluster c at decision epoch t has a budget $\beta_{c,t}^{avail} \in \{1, 2, \dots, |T_s|\}$. The budget $\beta_{c,t}^{avail}$ of cluster c represents the total amount of white spaces available for cluster c at decision epoch t. The budget changes with the amount of white spaces available to both clusterhead and each of its member nodes and it is calculated as follows:

$$w_t^i = \sum_{t_s=1}^{|T_s|} \bigcup_{m=1}^{|M|} w_{m,t_s}^i \tag{2}$$

$$\text{s.t. } \Lambda_{m,t_s}^i = 1$$

The total amount of white spaces w_t^i for SU i at decision epoch t is the summation of all the available white spaces from time slots 1 to $|T_s|$. Hence, the total availability of white spaces for SU i at each time slot t_s over all the channels in a CRN at decision epoch t is as follows:

$$\beta_{c,t}^{avail} = \sum_{t_s=1}^{|T_s|} \bigcap_{i=1}^{|\mathcal{I}|} w_{t_s}^i \tag{3}$$

As an illustrative example for white space computation, please refer to Fig. 1 (a clustered network) and Fig. 2 (white spaces). In Fig. 1, a cluster is formed in a CRN based on the number of available common channel(s) in the cluster. In our work, as the main objective of the cluster size adjustment scheme is to maximize the use of available resources in CRNs, it is vital to choose a node with the highest number of available common channels as the clusterhead in order to maximize the allocation of the resources to its member nodes. To facilitate a successful transmission, at least a single common channel must be available among the nodes in a cluster so that a clusterhead can broadcast information to its member nodes. We assume that SUs are randomly distributed, with $M = \{1, 2, 3, 4\}$ channels, and each PU transmits in a single channel *m* with transmission range r_m, so a SU is either located within or outside a PU's transmission range. Cluster C_1 consists of clusterhead CH_1 as well as member nodes SU_1 and SU_2. Cluster C_1 has two available common channels $m = \{3, 4\}$, where channel 3 is available due to the *lack* of PU's activities from $PUBS_3$, while channel 4 is available at all times as the SU_1 and SU_2 are located out of the transmission range r_4 of channel 4 used $PUBS_4$. Note that channel 1 is unavailable due to the presence of PU's activities from $PUBS_1$ as SU_1 is located within its transmission range r_1, and channel 2 is unavailable due to the presence of PU's activities from $PUBS_2$ as CH_1 and SU_2 are located within its transmission range r_2.

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

Fig. 2 shows a clusterhead and its member nodes in CRN operating environment with $|T_s| = 4$ time slots and $|M| = 4$ channels. From Fig. 1, each member node has at least one channel that is always available. Using (2), at the node level, SU_1 at decision epochs $t = 1$ and $t = 2$ has a total amount of white spaces $w_1^1 = 3$ and $w_2^1 = 4$, respectively. Using (3), at the cluster level, cluster C_1 at decision epochs $t = 1$ and $t = 2$ has budget $\beta_{c,1}^{avail} = 2$ and $\beta_{c,2}^{avail} = 4$, respectively. The white spaces at decision epoch $t = 1$ are relatively less due to the unavailability of the channels at time slot $t_s = 2$ for SU_1, SU_2 and CH_1, and at time slot $t_s = 3$ for SU_1.

In summary, the presence of PUs' activity ($\Lambda_{m,t_s+1} = 0$) reduces the budget value, $\beta_{c,t}^{avail}$ forcing clusterhead CH_c to experience a smaller cluster size as it has lesser white spaces to distribute to its member node(s) $MN^{i,h}$. Conversely, the absence of PUs' activity ($\Lambda_{m,t_s+1} = 1$) increases the budget $\beta_{c,t}^{avail}$, thus enabling clusterhead CH_c to have a bigger cluster size since it has more white spaces to distribute to its member node(s) $MN^{i,h}$. The cluster size of cluster c at decision epoch t, $Z_{c,t}$ is the summation of all the member nodes of various hops in its cluster as follows:

$$Z_{c,t} = \sum_{h} \sum_{i} MN^{i,h} \tag{4}$$

C. Cluster Size Adjustment Scheme

In our cluster size adjustment scheme, we consider a clusterhead CH_c in cluster c at decision epoch t has budget $\beta_{c,t}^{avail}$, which is the total amount of transmission opportunities in a CRN. A clusterhead CH_c distributes its budget $\beta_{c,t}^{avail}$ in the form of tokens $\mathcal{T}$ (i.e., white spaces), to its member nodes $MN^{i,h}$. Each member node with token(s) is given the opportunity to transmit its data packets. Clusterhead CH_c considers a token is utilized once it has granted it to its member node for data packets transmission purposes. Note that the further a node is away from its clusterhead, the higher the number of tokens are needed for data packet transmission. For instance, a one-hop member node $MN^{i,1}$ needs only one $\mathcal{T}$ to transmit its data packet, while member node $MN^{i,h}$ needs h $\mathcal{T}$ to perform the same task. We assume that a member node $MN^{i,h}$ only knows its one-hop upstream (or parent) node. For a node to join the cluster as a member, the parent member node $MN^{i,h}$, which is h-hop away from the clusterhead, needs an additional $h+1$ $\mathcal{T}$ from CH_c in order for its downstream node to transmit its data packets successfully. We denote member node $MN^{i,h}$ successful data packet transmission as $\beta_{i,t}^{used}$.

As an illustrative example for budget $\beta_{c,t}^{avail}$ allocation as shown in Fig. 3, we re-consider Fig. 1 and Fig. 2. At decision epoch $t = 2$, in cluster C_1, clusterhead CH_1 has a budget value $\beta_{c,2}^{avail} = 4$. At the start of decision epoch $t = 2$, SU_1 and SU_2 request to join cluster c and so they request 1 and 3 $\mathcal{T}$ (i.e., $\beta_{1,2}^{request} = 1$ and $\beta_{2,2}^{request} = 3$), respectively. As mentioned earlier, a one-hop node requires one $\mathcal{T}$ for data transmission. However, note that SU_2 requests an additional two $\mathcal{T}$s as it intends to add SU_3, which is its downstream two-hop node. Suppose SU_1 is a legitimate member node but not SU_2, and clusterhead CH_1 not having prior knowledge of their behavior, grants tokens $\mathcal{T}$ to SU_1 and SU_2 (i.e., $\beta_{1,2}^{granted} = 1$

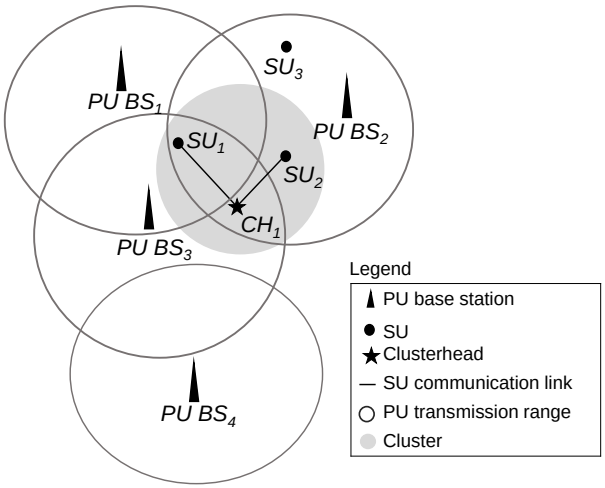

Fig. 1: A scenario of a clustered network with *four* PU base stations — $PUBS_1$ – $PUBS_4$ own channels 1 – 4, respectively. A cluster has a clusterhead CH_1, and two member nodes SU_1 and SU_2; they have common channels 3 and 4 — channel 3 is underutilized and channel 4 is out of the transmission range. SU_3 is a non-member of the cluster.

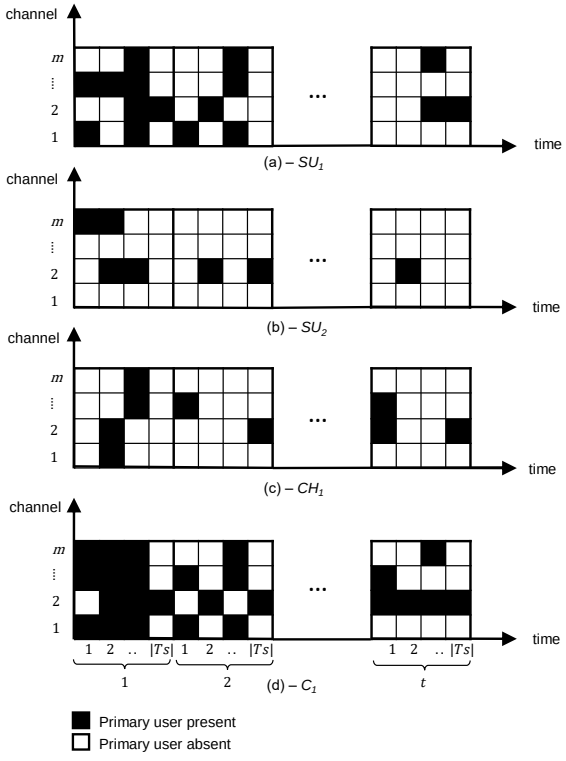

Fig. 2: A sample of white spaces at SUs, clusterhead CH_1, and cluster C_1.

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

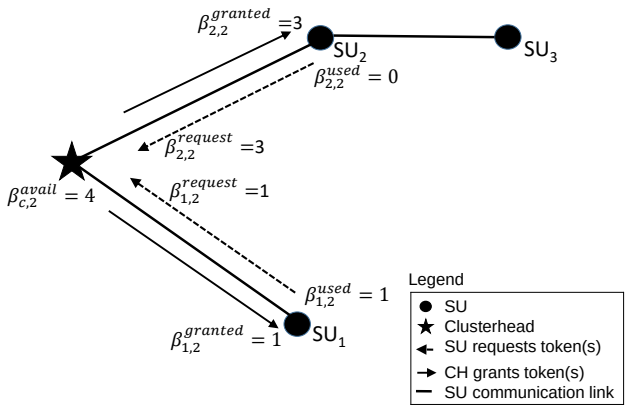

Fig. 3: A budget scenario for clusterhead CH_1 in cluster C_1 at decision epoch $t = 2$.

and $\beta_{2,2}^{granted} = 3$), respectively. After data transmission (i.e., at the end of decision epoch $t = 2$), SARL clusterhead CH_1 is able to observe the amount of $\mathcal{T}$ used by its member nodes. In our example, given that SU_2 is malicious, the amount of token wasted is $\beta_{2,2}^{used} = 0$ since it does not utilize the given token to help clusterhead CH_1 to add a downstream node as well as to join itself as a one-hop member node. Hence, from the clusterhead CH_1 perspective, the total amount of budget used is $\beta_{c,2}^{used} = 1$.

D. Attack Model

In our attack model, malicious SU i aims to minimize the token utilization (or $\beta_{c,t}^{used}$) without being detected by the clusterhead, and has its own perceived performance indicator D_t^i denoting how well it has performed in the attack after each decision epoch t. The performance indicator value, D_t^i is the perceived clusterhead trust value of the malicious SU i, and it is denoted as follows:

$$D_t^i = \frac{\beta_{i,t}^{request}}{\beta_{i,t}^{granted}} \tag{5}$$

A malicious SU's behavior is defined by the attack probability and intensity. Its possible actions are denoted as $a_t^i = (P_t^i,\ I_t^i) \in A$, where $0.1 \leq P_t^i \leq 0.9$ is the attack probability and $0.1 \leq I_t^i \leq 1$ is the attack intensity. The attack probability and intensity represent the attack frequency and strength, respectively. The attack probability $P_t^i = 1$ is not considered in our scenarios so as to provide some randomness in the attack. Note that the effects of the malicious SU's attack to cluster size vary as time goes by since the SU is capable of attacking with various possible actions to avoid being detected. An attack causes the tokens granted $\beta_{i,t}^{granted}$ to a malicious SU to become underutilized and its action not being detected by the clusterhead. However, when a clusterhead detects an attack, the malicious SU may be punished with a reduced trust value, and risks being removed from its cluster. The knowledge of its previous attack, particularly the effects of attack on cluster scalability is used as a decision making factor for the next attack.

III. Problem Formulation

When a clusterhead CH_c has a certain budget $\beta_{c,t}^{avail}$ based on the availability of white spaces at decision epoch t, it attempts to distribute the budget in the form of tokens $\mathcal{T}$ to its member nodes so as to maximize the budget utilization, leading to larger cluster size and hence higher cluster scalability. However, due to the hostile environment in CRNs, a member node $MN^{i,h}$ may request more tokens from clusterhead CH_c with the intention of admitting new nodes to cluster c but underutilizes them. Hence, at the node level, the token utilization of a member node $MN^{i,h}$ is the amount of white spaces used, and it can be measured by data packet transmission activities. The unused amount of the budget for member node $MN^{i,h}$, namely $\beta_{i,t}^{unused}$ can be denoted as follows:

$$\beta_{i,t}^{unused} = \beta_{i,t}^{granted} - \beta_{i,t}^{used} \tag{6}$$

As clusterhead CH_c's objective is to maximise the cluster's budget utilization, it aims to minimize $\beta_{c,t}^{unused}$ for the cluster. The unused budget $\beta_{c,t}^{unused}$ is the summation of all the unused budget of its members $\beta_{i,t}^{unused}$. This problem can be formulated as follows:

$$\beta_{c,t}^{unused} = \min(\beta_{c,t}^{avail} - (\sum_{i=1} \beta_{i,t}^{granted} - \sum_{i=1} \beta_{i,t}^{used})) \tag{7}$$
$$\text{s.t. } \beta_{c,t}^{avail} > 0$$

On the contrary, the malicious SU i's objective is to minimize or underutilize the given tokens $\mathcal{T}$. In other words, it aims to maximize $\beta_{i,t}^{unused}$. This problem can be formulated as follows:

$$\beta_{i,t}^{unused} = \max(\beta_{i,t}^{granted} - \beta_{i,t}^{used}) \tag{8}$$
$$\text{s.t. } \beta_{c,t}^{avail} > 0$$

IV. RL Algorithms

In the SARL approach, which is the single-agent approach, each individual decision maker (or agent) learns from the operating environment independently; while in the MARL approach, which is the multi-agent approach, each agent learns from the operating environment and collaborates with its neighboring agents via exchanging local payoffs among themselves, so that it can evaluate and coordinate its action as part of the joint action, which is the action selected by neighboring agents. In our work, SARL is embedded in both clusterheads and malicious SUs while MARL is embedded in the malicious SUs only. The SARL algorithm maximizes the local Q-value; while the MARL algorithm maximizes the global Q-value (or the sum of the local Q-values of all agents).

A. SARL Algorithm

The SARL algorithm embedded in each agent i is shown in Algorithm 1. An agent i observes *state* $s_t^i \in S$, which is the local decision making factors from the dynamic operating environment and selects an *action* $a_t^i \in A$ at decision epoch t. Either an *exploitation* or an *exploration* action is selected.

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

ALGORITHM 1: SARL

procedure
  Observe current state s_t^i
  if exploration **then**
    Select a random action a_t^i
  else
    Select an optimal action $a_t^{i,*}$ using Eq. (9)
  end if
  Receive delayed reward $r_{t+1}^i(s_{t+1}^i, a_{t+1}^i)$
  Update Q-value $Q_{t+1}^i(s_t^i, a_t^i)$ using Eq. (10)
end procedure

The exploitation action is a greedy action with the maximum Q-value, which represents the appropriateness of a state-action pair, as follows:

$$a_t^{i,*} = \underset{a \in A}{\operatorname{argmax}}\, Q_t^i(s_t^i, a) \tag{9}$$

The exploration action is a random action selected to update the Q-values of non-greedy actions so that better actions may be discovered. Next, the agent i receives *delayed reward* $r_{t+1}^i(s_{t+1}^i, a_{t+1}^i)$, which represents the positive or negative effect from its local operating environment at the next decision epoch $t+1$ [8]. As the decision epoch progresses $t = 1, 2, \ldots$, an agent i explores all state-action pairs (s_t^i, a_t^i) and updates their respective Q-values $Q_t^i(s_t^i, a_t^i)$ using Q-function as follows:

$$\begin{aligned} Q_{t+1}^i(s_t^i, a_t^i) \leftarrow (1-\alpha) Q_t^i(s_t^i, a_t^i) \\ + \alpha[r_{t+1}^i(s_{t+1}^i) + \gamma \max_{a \in A} Q_t^i(s_{t+1}^i, a)] \end{aligned} \tag{10}$$

where $0 < \alpha < 1$ represents the learning rate, and $0 < \gamma < 1$ represents the discount factor.

The SARL algorithm embedded in each agent i may also be stateless; the *state* s_t^i representation can be omitted while still achieving its goal i.e., the *delayed reward* r_{t+1}^i does not depend on the state but action only. In a stateless model an agent i's state never changes throughout its operation and it is denoted as follows:

$$\begin{aligned} Q_{t+1}^i(a_t^i) \leftarrow (1-\alpha) Q_t^i(a_t^i) \\ + \alpha[r_{t+1}^i(a_{t+1}^i) + \gamma \max_{a \in A} Q_t^i(a)] \end{aligned} \tag{11}$$

where $0 < \alpha < 1$ represents the learning rate, and $0 < \gamma < 1$ represents the discount factor.

B. MARL Algorithm

The MARL algorithm as shown in Algorithm 2, extends the SARL algorithm with local payoff exchange, is embedded in each agent i. In a multi-agent setting, multiple agents exchange future rewards among themselves at decision epoch t, and update their Q-values simultaneously at decision epoch $t+1$. In Fig. 4(a), at decision epoch t, an agent i receives future rewards $\max_{a^j \in A} Q_t^j(s_t^j, a^j)$, which are locally maximized, from its neighboring agent $j \in J^i$, where J^i is a set of agent i's neighbors. Towards the end of decision epoch t, the agent i would have received future rewards $\max_{a^j \in A} Q_t^j(s_t^j, a^j)$ from its neighboring agents J^i. Note that $\max_{a^j \in A} Q_t^j(s_t^j, a^j)$ represents a future reward based on state s_t^j, and it is received from neighboring agents J^i at decision epoch t. In Fig. 4(b), at decision epoch $t+1$, the agent i receives delayed reward $r_{t+1}^i(s_{t+1}^i)$ and learns the best possible action for a particular state s_t^j, which is part of the joint action, using Equation (12) in order to achieve an optimal global reward at decision epoch $t+1$. This allows an agent to evaluate its own action while taking into account the average future rewards of its neighboring agents, where the future reward of each neighbor j is weighted by $\eta^{i,j} = 1/|J^i|$. Equation (12) has been applied in a number of papers, including [26]–[28].

ALGORITHM 2: MARL

procedure
  Observe current state s_t^i
  if exploration **then**
    Select a random action a_t^i
  else
    Select an optimal action $a_t^{i,*}$ using Eq. (9)
  end if
  Receive shared information from neighboring agent $j \in J^i$
  Receive delayed reward $r_{t+1}^i(s_{t+1}^i)$
  Calculate combined information from neighboring agents and itself
  Update Q-value $Q_{t+1}^i(s_t^i, a_t^i)$ using Eq. (12)
end procedure

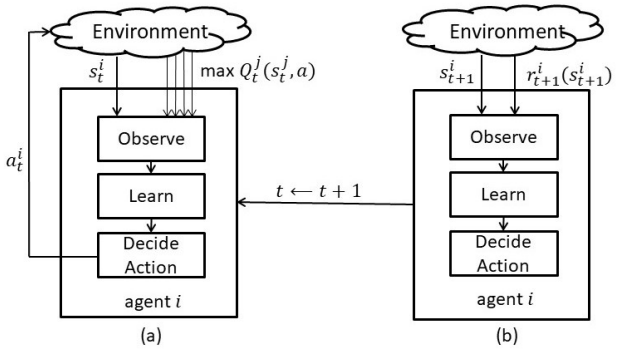

Fig. 4: An abstract model for agent i at (a) decision epoch t, and (b) decision epoch $t+1$

$$\begin{aligned} Q_{t+1}^i(s_t^i, a_t^i) \leftarrow (1-\alpha) Q_t^i(s_t^i, a_t^i) \\ + \alpha[r_{t+1}^i(s_{t+1}^i) + \gamma \sum_{j \in J^i} \eta^{i,j} \max_{a^j \in A} Q_t^j(s_t^j, a^j)] \end{aligned} \tag{12}$$

Generally speaking, there are four main differences between MARL and another learning mechanism, namely non-cooperative game (e.g., prisoners' dilemma), where decision makers select actions independently. Firstly, MARL enables decision makers to select actions based on their own, and their respective neighbors, actions and payoffs in order to achieve optimal joint action, contributing to optimal network-wide performance [29], while non-cooperative game enables

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

decision makers to select actions based on their own action and payoff only in order to improve local network performance [30]. Secondly, MARL requires local information to compute optimal joint action, while non-cooperative game requires a complete set of global information to compute Nash equilibrium that provides the optimal local action [31], [32]. Thirdly, MARL learns as time goes by to maximize long-term rewards, while non-cooperative game computes Nash equilibrium for each episode (or time window). Fourthly, MARL allows the agents' behavior to change as time goes by in a dynamic operating environment, while non-cooperative game assumes each decision maker to react rationally.

V. RL-BASED ATTACK MODEL AND TRUST MODEL FOR CLUSTER SIZE ADJUSTMENT

This section discusses the two separate RL-based models for both the attacker and the clusterhead. In the RL-based attack models for the attackers the state s_t^i represents the performance indicator D_t^i as seen in (13) and its action a_t^i is the combination of two sub-actions, namely the probability of attack P_t^i and the intensity of attack I_t^i. In the RL-based trust model for the clusterhead, it does not have a state (i.e., at each decision epoch, its state never changes) and yet it can still achieve its goal by relying on the action it takes as seen in (11). We have summarized the RL components for the attacker and the clusterhead in Tables IV and V.

A. Attack Model Embedded in Each Malicious SU

In addition to the traditional random and deterministic attacks, where the latter employs with certain attack probability and intensity, the malicious SUs can also leverage on SARL and MARL to launch attacks. In the SARL approach, each malicious SU i launches attacks independently with the objective of reducing cluster scalability by not cooperating with the clusterhead to add new nodes when it has been granted with the requested amount of tokens $\mathcal{T}$. Specifically, it maximizes the unused budget $\beta_{i,t}^{unused}$ (see 6) and avoids being removed from the cluster by its clusterhead, which either increases or reduces its trust value. In this intelligent attack, the malicious SU i learns the operating environment by observing its current performance indicator $D_t^i \in [0,1]$, which is calculated using (5). The performance indicator D_t^i is uniformly partitioned into four sub-ranges as seen in (13), and so the *state* of each malicious SU can be represented as follows:

$$s_t^i = \begin{cases} 1\ (poor) & 0 \le D_t^i < 0.25, \\ 2\ (average) & 0.25 \le D_t^i < 0.5, \\ 3\ (good) & 0.5 \le D_t^i < 0.75, \\ 4\ (best) & 0.75 \le D_t^i \le 1. \end{cases} \tag{13}$$

We assume that each SU i has its initial performance indicator value of 0.25. The action a_t^i is the combination of two sub-actions, namely attack probability, $0 \le P_t^i \le 0.9$ and attack intensity $0 \le I_t^i \le 1$. The attack intensity is measured by the amount of tokens used by the malicious SU. Lower intensity indicates higher white space utilization

TABLE IV: RL model embedded in each attacker

State	$s_t^i \in S = (1, 2, \dots 4)$ represents the performance indicator for attacker i at decision epoch t. States $s_t^i = 1$ and $s_t^i = 4$ indicate that the attacker has the worst and the best performance, respectively
Action	$a_t^i \in A = (P_t^i, I_t^i)$ represents the sub-actions of the probability of attack P_t^i and the intensity of attack I_t^i, where $0.1 \le P_t^i \le 0.9$ and $0.1 \le I_t^i \le 1$
Reward	$r_t^i = \beta_{i,t}^{unused}$ represents the unused white spaces

TABLE V: RL model for cluster size adjustment embedded in a clusterhead

Action	$a_t^i \in A = \{a_t^i \mid \max Q_t^i(a_t^i)\}$ where $i = \{1, 2, \dots, n\}$, represents set of one-hop member nodes in a cluster —node with the highest Q-value is chosen, followed by a node with the second highest Q-value, and so on.
Reward	$r_t^i = \beta_{i,t}^{used}$ represents the total amount of used white spaces by the chosen member nodes at decision epoch t

by the attacker i.e., lower ($\beta_{i,t}^{unused}$). An action a_t^i of (P_t^i=0.1, I_t^i=0.1) taken by a SU indicates an attack with the lowest attack probability and intensity, while an action a_t^i of (P_t^i=0.9, I_t^i =1) taken by a SU indicates an attack with the highest probability and intensity.

The *delayed reward* of a malicious SU i is $r_t^i = \beta_{i,t}^{unused}$, which consists of the unused white spaces. The unused white spaces are caused by the unutilized resources not used for data packet transmission and/or the help to the clusterhead to add member nodes.

The SARL algorithm embedded in each SU i is shown in Algorithm 1. Note that each SU i is a potential attacker and it leverages on the SARL model to learn the best combination of sub-actions given a state so as to avoid being detected and removed by the clusterhead. After carrying out the attack at decision epoch t, the malicious SU updates the state-action pair (s_t^i,a_t^i) in its Q-table using (10)

B. Trust Model Embedded in a Clusterhead for Cluster Size Adjustment

Through the token allocation process, at each decision epoch t, clusterhead CH_c tracks the behavior of its member nodes $MN^{i,1}$ (i.e., one-hop downstream nodes) using Q-value. In the SARL approach, clusterhead CH_c select its legitimate member nodes by selecting the highest $Q_t^i(a_t^i)$ values. Member node $MN^{i,1}$ is considered legitimate when it follows the clustering rules: a) adds node(s) based on the requested number of tokens, and b) obeys to drop itself and its downstream node(s) from the cluster when the requested number of tokens is not granted.

The clusterhead applies a trust model to learn the behavior of its member nodes in order to distribute the tokens to them so as to efficiently maximize its cluster size. Denote $a_t^i \in A = \{a_t^i \mid \max Q_t^i(a_t^i)\}$ where $i = \{1, 2, \dots, n\}$, is a set of one-hop member nodes with the highest Q values in a cluster. Upon action selection, the delayed reward is $r_t^i = \beta_{c,t}^{used}$, which is the used white space after the each chosen node has added new

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

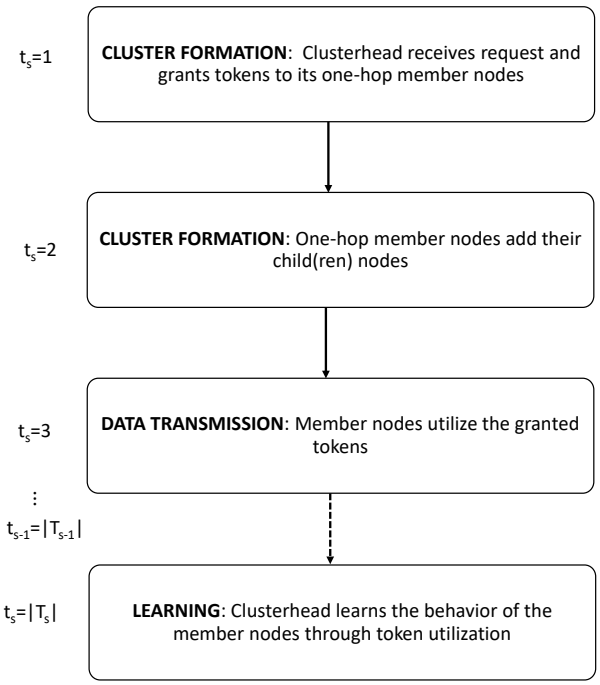

Fig. 5: A flowchart that depicts the various phases of a cluster size adjustment scheme at decision epoch t that consists of $t_s = |T_s|$ time slots.

nodes and/or transmitted its data packets and/or relay packets for its downstream nodes. The clusterhead then updates its member nodes' Q-values accordingly.

Fig. 5 shows the various phases of a cluster size adjustment scheme in the form of a flowchart. Each decision epoch t consists of $t_s = |T_s|$ time slots. During the cluster formation, at time slot $t_s = 1$, the clusterhead gathers information and forms a cluster based on its neighboring nodes Q-values. During this phase, the clusterhead grants tokens to its one-hop member node —a node with the highest Q-value is granted the most tokens, followed by a node with the second highest Q-value and so on, and nodes with lower Q-values may be granted with tokens, subject to the availability of the remaining tokens. At time slot $t_s = 2$, the one-hop member nodes add their child(ren) nodes based on the granted tokens. At time slot $t_s = 3$, data transmission begins and ends at $t_s = |T_{s-1}|$. At time slot $t_s = |T_s|$, the clusterhead learns about the legitimate level of its member nodes – a member node has a higher legitimate level if it has utilized all the granted tokens during data transmission. In our scenario, we assume that the demand for nodes to join a cluster is higher than the available tokens. Hence, there is no issue of token wastage from the clusterhead's perspective. As an illustrative example, during the cluster formation at decision epoch t, a clusterhead has 10 tokens and has learnt knowledge (i.e., Q-value) of the nodes in the network. Nodes A, B and C with Q-values of 0.4, 0.3 and 0.3 respectively, are granted at most 4, 3, and 3 $\mathcal{T}$ tokens, respectively. However, if nodes A, B and C requested only 3, 1 and 1 $\mathcal{T}$ tokens respectively, the remaining tokens are then given to other new nodes in the network to avoid wastage.

VI. Performance Evaluation, Results and Discussion

There are two main entities of interest in our simulation, namely SUs (potential attackers) and clusterhead. An intelligent attacker adopts SARL to avoid being detected when launching independent attacks, while an intelligent clusterhead adopts SARL to detect attackers, remove them from its cluster, and grant its budget to legitimate SUs so as to increase the cluster size.

A. Simulation Overview

1) Summary of Tasks: In our simulation work, we perform the following tasks. Firstly, we investigate the nature of the non-SARL attacks, namely random and deterministic, to discover their similarities or differences (see Section VI-B). Secondly, based on the results about their characteristics, we establish four scenarios of attack with non-SARL or SARL embedded in clusterhead and/or attackers, and use the results as baselines (see Section VI-C1). Thirdly, we establish a rule-based SARL for clusterhead to counter SARL attacks and show its further improvement to the cluster size ratio (see Section VI-C2). Finally, we investigate the MARL attackers' performance on a SARL clusterhead, and discuss the effectiveness of MARL attacks in a dynamic operating environment (see Section VI-C3).

2) Assumptions: We consider at each decision epoch t, clusterhead CH_c in cluster c attempts to distribute its budget in the form of tokens $\mathcal{T}$ to its neighboring nodes in order to maximize the budget utilization so as to improve network scalability.

Clusterhead CH_c aims to maximize cluster size based on budget $\beta_{c,t}^{avail}$. To implement this scenario, SARL algorithm is embedded in clusterhead CH_c so that, at the end of each decision epoch t (i.e., after data packet transmission), CH_c learns its member nodes' behavior through Q-value updates in order to identify more reliable nodes (i.e., with higher trust values) as time goes by. We assume that clusterhead CH_c is legitimate at all times and its cluster size at each decision epoch t is influenced by the given budget based on the amount of white spaces, and there are always SUs with certain attack probability P_t^i and intensity I_t^i that want to join the cluster. Throughout decision epoch T, we also assume that a cluster c has a maximum average attack probability $\mathbb{P}_c$ and intensity $\mathbb{I}_c$, which are both derived from the initial mean of the attack probabilities P_0^i and intensities I_0^i of all its attackers, respectively, and they are shown in (14) and (15). Henceforth, the attack probability and intensity discussed in our article refer to the cluster's attack probability and intensity, namely $\mathbb{P}_c$, and $\mathbb{I}_c$, unless otherwise mentioned.

$$\mathbb{P}_c = \frac{\sum_{i=1}^{|\mathcal{I}|} P_0^i}{|\mathcal{I}|} \tag{14}$$

$$\mathbb{I}_c = \frac{\sum_{i=1}^{|\mathcal{I}|} I_0^i}{|\mathcal{I}|} \tag{15}$$

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

3) Performance Metric: In our work, we measure the attack performance based on cluster size ratio that compares cluster sizes before and after an attack at the end of each decision epoch t as shown in (16). Higher cluster size ratio is desirable as it indicates greater network scalability. Table VI shows the simulation parameters. For simplicity, we consider only two PUs with the same transition probability matrix, namely $\mathcal{P} = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$. This is because the number of PUs in CRN does not affect the budget value as seen in (2).

$$\mathbb{R}_{c,t} = \frac{Z_{c,t_s=|T_s|}}{Z_{c,t_s=1}} \tag{16}$$

where $Z_{c,t_s=|T_s|}$ is the cluster size measured at the end of a decision epoch t after an attack has taken place, and $Z_{c,t_s=1}$ is the cluster size measured at the start of a decision epoch t when the clusterhead has granted its tokens to its member nodes.

B. Preliminaries: Analysis of non-SARL Attacks - Random and Deterministic Attacks

In this section, we analyze the effects of non-SARL attacks, namely random and deterministic attacks on the cluster size ratio.

We consider a scenario where both SUs and clusterhead do not adopt SARL, and the attackers attack in cluster c with an attack probability $0.1 \leq \mathbb{P}_c \leq 0.9$. In a random attack, an attacker attacks with a random intensity that ranges from $0 \leq I_t^i \leq 0.9$, while in a deterministic attack, an attacker attacks with an attack intensity I_t^i. Fig. 6 shows that the cluster size ratio decreases as the attack probability $\mathbb{P}_c$ and intensity $\mathbb{I}_c$ increase.

In our simulation, given an attack probability $\mathbb{P}_c$, both random and deterministic (with intensity $\mathbb{I}_c = 1$) attacks yield similar cluster size ratio, and their respective 95% confidence interval shown in Fig. 7, establish that both random and deterministic (with intensity $\mathbb{I}_c = 1$) attacks are similar in nature, which have similar maximum and minimum ratios. Hence, both random and deterministic (with intensity $\mathbb{I}_c = 1$) attacks are considered to be equivalent. Henceforth, deterministic (with intensity $\mathbb{I}_c = 1$) attacks are also referred as random attacks.

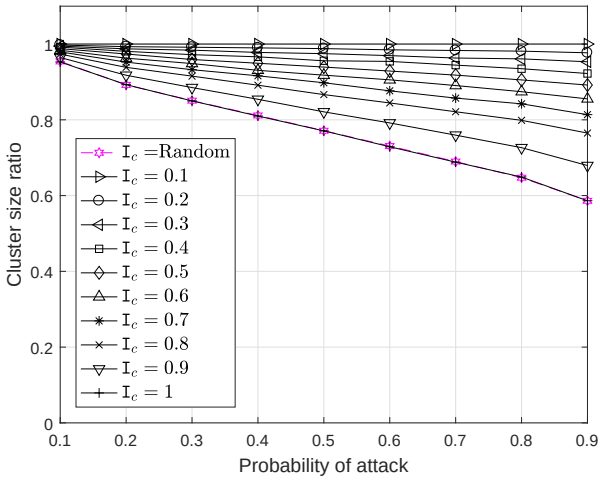

Fig. 6: Cluster size ratio reduces with increasing attack probability $\mathbb{P}_c$ and intensity $\mathbb{I}_c$. *Random and deterministic ($\mathbb{I}_c = 1$) attacks cause the most detrimental effects to a cluster.*

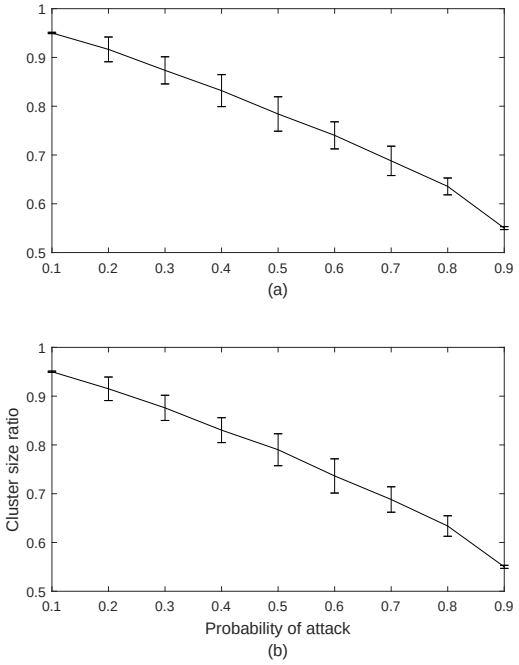

Fig. 7: Cluster size ratio for non-SARL attack algorithms (with confidence interval 95%). (a) Random attack (b) Deterministic attack ($\mathbb{I}_c = 1$). *Both attacks have similar maximum and minimum ratios.*

C. Simulation Setup and Analysis

The following *three* cases, namely Case I, Case II, and Case III establish the conditions as proposed in Section I-D. In our work, the main objective of the SARL malicious SUs is to waste the tokens granted to them and yet remain undetected, while the SARL clusterhead aims to learn its member nodes behavior and remove the malicious ones from the cluster. By wasting the tokens, the malicious SUs reduce the cluster size.

1) Case I: We investigate the effectiveness of using SARL approach in attacks by considering the following *four* scenarios: in Scenario (a), both attackers and clusterhead adopt non-SARL approaches; in Scenario (b), attackers adopt non-SARL approach and the clusterhead adopts the SARL approach; in Scenario (c) attackers adopt SARL approach and the clusterhead adopts non-SARL approach, and in Scenario (d), both attackers and clusterhead adopt the SARL approaches. Scenarios (a) – (c) are used as baselines while Scenario (d) refers to Case I of our contribution stated in Section I-D. Table VII provides a summary of these scenarios.

Fig. 8 presents the results of the four attack scenarios. The four sub-graphs have similar trends with cluster size ratio decreasing when attack probability $\mathbb{P}_c$ and intensity $\mathbb{I}_c$ increase. Figs. 8(a) and (c) show that when the non-SARL

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

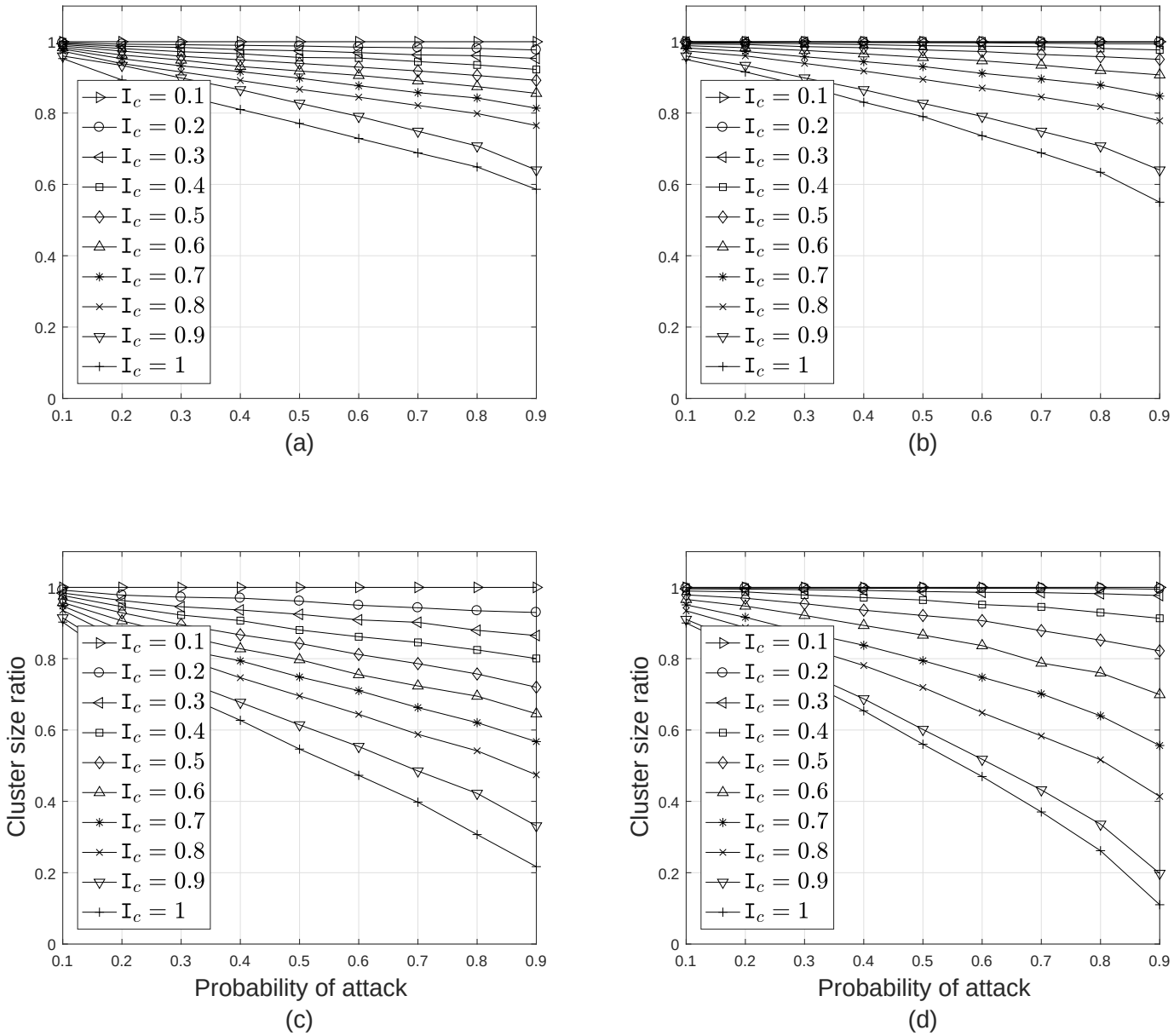

Fig. 8: Cluster scalability vs. varying attack intensity: Scenario (a), both attackers and clusterhead use non-SARL approaches; Scenario (b), attackers use non-SARL approach and the clusterhead uses the SARL approach; Scenario (c), attackers use SARL approach and the clusterhead uses non-SARL approach; Scenario (d), both attackers and clusterhead use the SARL approaches. Cluster size ratio decreases with increasing attack probability $\mathbb{P}_c$. *SARL attackers achieve the lowest cluster size ratio* (which also indicates lower cluster scalability) as seen in Scenario (c) and (d).

clusterhead unequivocally grants tokens to its member nodes, the SARL malicious member nodes learnt its clusterhead characteristics (Scenario (c)), and increase their attack capabilities, resulting in a cluster size ratio drop to 37%. Conversely, Figs. 8(b) and (d) show that when a clusterhead adopts the SARL approach to detect and remove its malicious nodes from the cluster, the SARL malicious member nodes learnt its clusterhead characteristics (Scenario (d)) and adjusted their attack capabilities to avoid being detected, resulting in a significant cluster size ratio drop to 44%. We conclude that SARL attackers cause a more detrimental effect to the cluster size ratio as compared to non-SARL attackers. In addition, the results in Figs. 8(a) and (d) show that when both attackers and clusterhead adopt SARL or non-SARL, the cluster size ratio reduced significantly to almost 48%, and this necessitates further work to enhance SARL clusterhead's learning process

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

TABLE VI: Simulation parameters

Notation	Description	Value
T	A decision epoch	10000
$\|M\|$	Number of primary users	2
$\mathcal{P}$	Transition probability matrix for each primary user	$\begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$
$\|\mathcal{I}\|$	Number of secondary users	50
$\beta_{c,t}^{avail}$	Available budget	≤ 15
α	Learning rate	0.5
γ	Discount rate	0.8

TABLE VII: Case I scenarios

Attackers \ Clusterhead	Non-SARL	SARL
Non-SARL	Scenario (a)	Scenario (b)
SARL	Scenario (c)	Scenario (d)

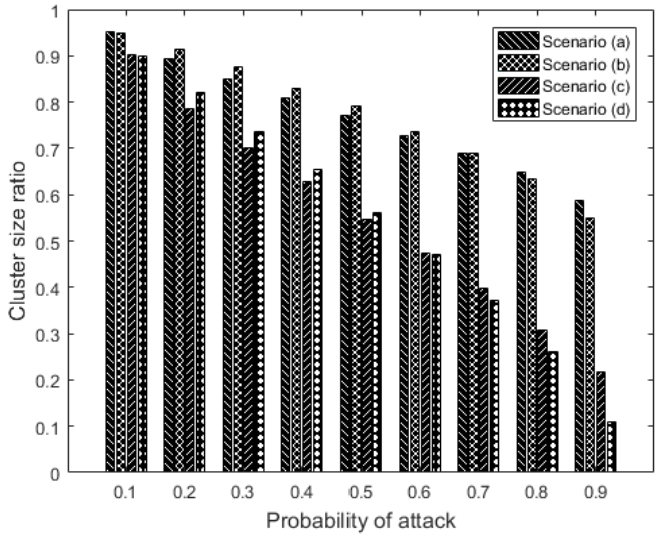

Fig. 9: Performance of attackers and clusterhead in variou scenarios — SARL clusterhead generally perform well i either non-SARL or SARL attacks as seen in Scenarios (b) an (d). However, *cluster size ratio decreases significantly whe attack probability* $\mathbb{P}_c \geq 0.6$, despite the adoption of SARL b the clusterhead as seen in Scenario (d).

in order to increase its performance in cluster scalability.

For simplicity sake, we focus on attack intensity $\mathbb{I}_c =$ in our subsequent simulations since all the four sub-graphs i Fig. 8 have similar trends – i.e., the cluster size ratio decrease when attack probability $\mathbb{P}_c$ and intensity $\mathbb{I}_c$ increase.

Fig. 9 and Fig. 10 compare the four attack scenario against the cluster size ratio. We find that non-SARL attacker performed poorly even as their attack probabilities increas as seen in Scenarios (a) and (b), where the cluster size ratio still remains at about 0.55 even when the attack probability $\mathbb{P}_c = 0.9$. This is because the attackers do not have the capability to observe their operating environment and learn from their previous attacks in order to launch a better subsequent attack. In Fig. 10, from the attackers' perspective, their attacks have caused the cluster size ratio to reduce by 37% and 44% in Scenarios (c) and (d), respectively. Note that even when the clusterhead adopted SARL to remove its malicious member nodes, the SARL attackers' performance improved significantly as the attack probability $\mathbb{P}_c$ increased. This is because when the attack probability $\mathbb{P}_c$ increased, the clusterhead failed to learn accurately and hence SARL clusterhead may not be efficient enough in an enviroment saturated with malicious SUs. As from the clusterhead's perspective when SARL is adopted, the cluster size ratio reduces by 4% and 11% in Scenarios (b) and (d), respectively. Note that SARL clusterhead adopts SARL to remove its malicious member nodes, the non-SARL attackers' do not make a significant impact to the cluster size even when the attack probability P increases. This is because they do not learn to avoid being detected by the clusterhead and hence in Scenario (c), the clusterhead is able to maintain a reasonable cluster size ratio as attack probability $\mathbb{P}_c$ increases.

From our investigation, we conclude that the SARL approach may not be effective when a cluster has a higher number of malicious SUs, as seen in Scenarios (b) and (d). When the environment becomes increasingly unstable, with $\mathbb{P}_c \geq 0.6$, the inaccurate learnt knowledge causes the SARL clusterhead to make wrong decisions. These observations warrant further investigations in order to provide a SARL clusterhead a better mechanism to detect malicious SUs of various probabilities. As for the marginal improvement on the ability of adopting SARL to detect malicious SUs as seen in Scenarios (a) compared to (b) and Scenarios (c) compared to (d), the SARL can be enhanced to take into account the instability of the operating environment. As such, we have addressed this issue in Conclusions and Future Work (Section VII). The following sub-section discusses a rule-based learning mechanism for the SARL clusterhead.

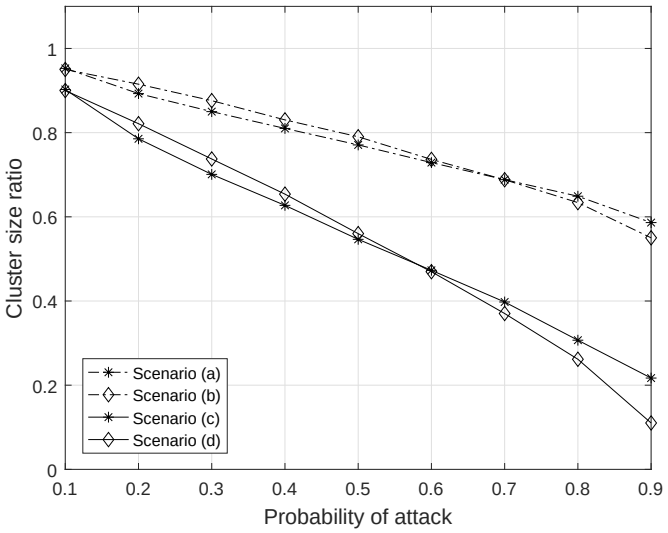

Fig. 10: *SARL-attackers achieve the lowest cluster size ratio* as seen in Scenario (c) and (d) (indicating poorer scalability).

2) Case II: We further fine-tuned the traditional SARL approach for the clusterhead by incorporating a rule-based approach, which is Case II alluded to in Section I-D. This

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

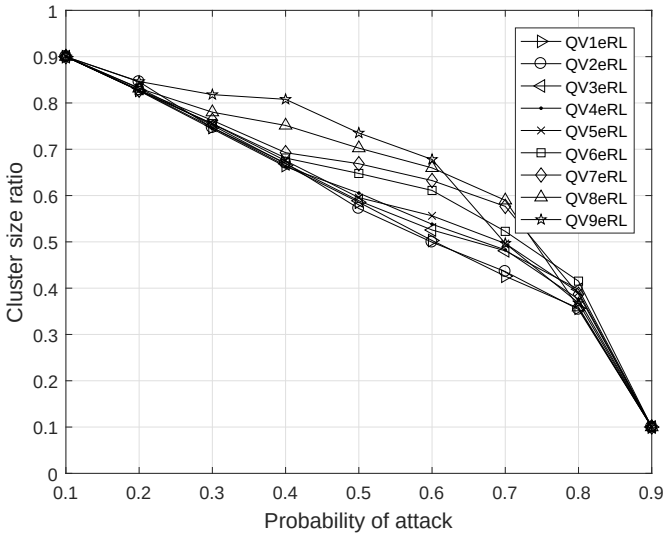

Fig. 11: Performance of rule-based SARL algorithm wit various thresholds: cluster size ratio decreases with increasing attack probability $\mathbb{P}_c$. *QV9eRL generally achieves the highest cluster size ratio when the attack probability* $\mathbb{P}_c \leq 0.6$. Higher cluster size ratio increases cluster scalability.

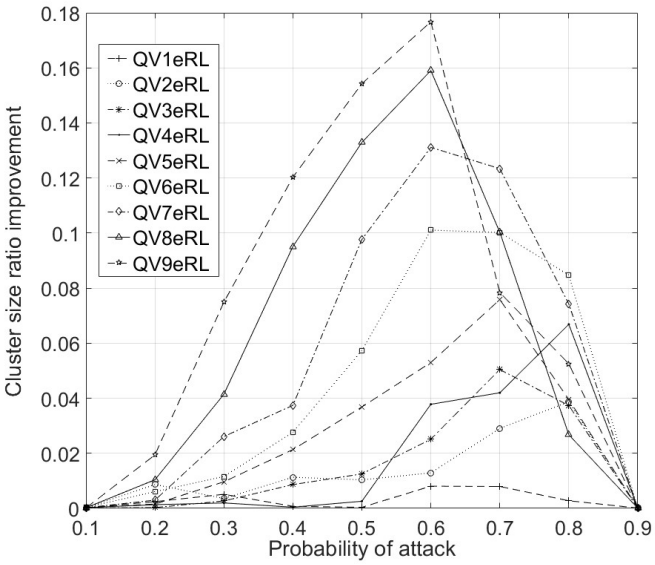

Fig. 12: Improvement of using rule-based SARL over traditional SARL approach. Rule-based SARL algorithms (QVeRL6 – QVeRL9) perform well when the attack probability $\mathbb{P}_c \leq 0.6$. *The increase in malicious activities (i.e.,* $\mathbb{P}_c \geq 0.7$*) causes the clusterhead to learn inaccurately resulting in the decrease in the cluster size ratio.*

approach allows nodes to join the cluster when they have met or exceeded certain Q-value, and to ensure *only* nodes of certain trustworthiness in terms of their Q-value are selected. For instance, QV1eRL and QV9eRL allow nodes, which have met or exceeded Q-values of 0.1 and 0.9, respectively to joi the cluster. When this rule-based method is applied to th nodes, the higher the threshold, the more trustworthy the node will be in a cluster, and this increases cluster scalability. W tested the rule-based SARL algorithm with varying threshold from 0.1 to 0.9 under the same parameter settings used by th non-SARL and the traditional SARL algorithms. The result have also shown that the cluster size ratio decreases when th attack probabilities $\mathbb{P}_c$ increase as is seen in Fig. 11. The rule based SARL, namely QV9eRL achieves the highest cluste size ratio since it only allows highly trustworthy nodes to joi the cluster (i.e., Q-values ≥ 0.9).

Fig. 12 shows the ratio improvement of the rule-base SARL algorithm with varying Q-value threshold. The rati improvement is the cluster size ratio difference between rule based SARL and traditional SARL approaches. Rule-base SARL, QV9eRL achieves the best performance as it increase the cluster size ratio up to 17% when the attack probabilit $\mathbb{P}_c \leq 0.6$. However, its performance drops as the attack probability $\mathbb{P}_c$ increases. While SARL clusterhead learns from its operating environment to increase cluster scalability, however, when the environment consists of malicious nodes with higher attack probability, the learnt knowledge may not be accurate.

3) Case III: We further increased the attackers' capability by adopting the MARL approach to launch attacks on the SARL clusterhead, as mentioned in our contribution in Section I-D. The MARL attackers in the cluster shared their learnt knowledge (i.e., Q-value) amongst themselves in order

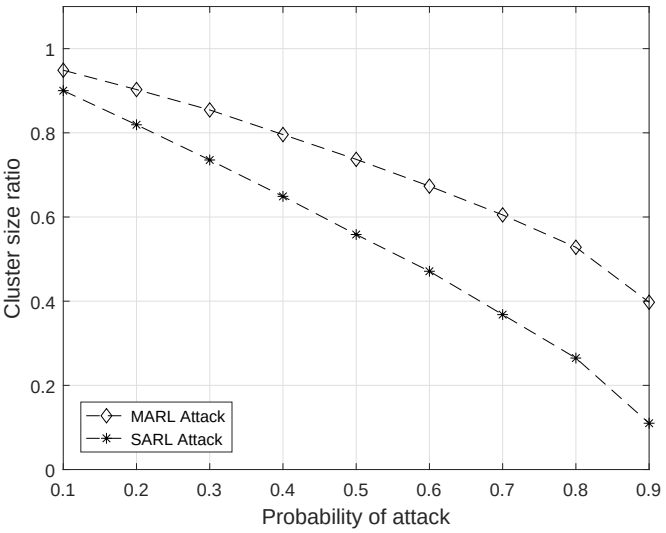

Fig. 13: Attacks performance using MARL (collaborative) vs. SARL (independent): *MARL attacks have shown to be less effective*; when compared with SARL attacks, MARL attacks' effectiveness reduces to 29% as the attack probability $\mathbb{P}_c$ increases since higher $\mathbb{P}_c$ increases the instability of the operating environment and this causes the learnt global knowledge to be inaccurate leading to a lesser effectiveness of the MARL attacks.

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

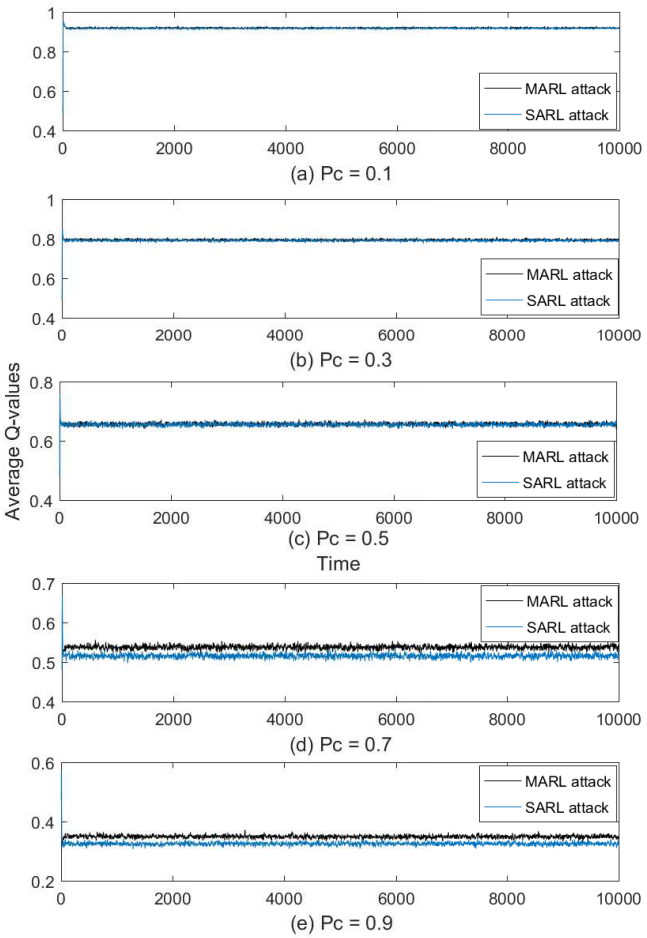

Fig. 14: Q-value comparisons for MARL and SARL attacks. Figs (a) – (e) show the average Q-values of all the nodes in a cluster with attack probability $\mathbb{P}_c$ of 0.1, 0.3, 0.5, 0.7 and 0.9, respectively. Similar trends are observed in other probabilities. *The MARL attackers have rather insignificant increase in the average Q-values*: 0.03, 0.03, 0.02, 0.22 and 0.24% for attack probability $\mathbb{P}_c$ equal to 0.1, 0.3, 0.5, 0.7 and 0.9, respectively.

to increase the effectiveness of their attacks. Fig. 13 shows the results of both MARL and SARL attackers launching attacks on its clusterhead with attack intensity $I_t^i = 1$. As the attack probability $\mathbb{P}_c$ increased in a cluster, both MARL and SARL attackers caused the cluster size ratio to be reduced to 17% and 29%, respectively.

Due to the MARL's characteristic of facilitating collaboration among SUs and sharing of learnt knowledge in order to have collective global knowledge, MARL attackers were expected to perform better than the SARL attackers [33]. However, our investigation proves otherwise.

In our simulated operating environment, which is dynamic in nature (i.e., changes in attack probability), the MARL attackers collaborate and share their learnt knowledge. When the learning process is done in an unstable operating environment, each attacker with different attack probability P_t^i will learn based on their own capacity; and as such the learnt knowledge will not be optimal causing the learnt global knowledge to be inaccurate [34]. Further analysis in Fig. 14 shows that the MARL attackers in a cluster generally have higher Q-values as their attacks are rather conservative as compared to SARL attackers, and hence they are regarded by the SARL clusterhead as less malicious. In addition, any increase in the number of malicious SUs to collaborate in the MARL operating environment will only cause further inaccuracy to the global knowledge, and hence the result in Fig. 13. In order for the attackers to take advantage of the MARL approaches, two factors are needed in the operating environment, namely stability and adaptation to the unpredicted behavior of other attackers [35].

D. Simulation Results and Discussions

Given the analysis in Section VI-C, we make comparisons amongst the three algorithms, namely non-SARL, traditional SARL and rule-based SARL (QV9eRL) algorithms. The non-RL scheme that employs Scenario (a) is used as a baseline, while both the traditional SARL and rule-based QV9eRL scheme employ Scenario (b). The results shown in Fig. 15 demonstrate that QV9eRL outperforms the non-SARL and the traditional SARL algorithms as the attack probability increases from $\mathbb{P}_c = 0.2$ to $\mathbb{P}_c = 0.8$. While the rule-based approach enables the SARL clusterhead to maximize its resources, it does not show any improvement in scenarios where attack probability $\mathbb{P}_c = 0.1$ and $\mathbb{P}_c = 0.9$. In a scenario where $\mathbb{P}_c = 0.1$, all the three algorithms perform equally well since the majority of the SUs are legitimate, and selecting the SUs randomly or based on learnt knowledge does not have much effect. From the SARL perspective, the absence of malicious SUs does not warrant the clusterhead to learn, hence all the three algorithms provide satisfactory performance for cluster size adjustment. On the other hand, in a scenario where attack probability $\mathbb{P}_c = 0.9$, all the algorithms fail to perform equally well as the majority of the SUs are all highly malicious in nature. In such a scenario, neither adopting SARL nor rule-based SARL (QV9eRL) algorithm may help increase the cluster size ratio as the malicious operating environment causes the clusterhead to learn inaccurately and hence, yield the lowest cluster size ratio. While this extreme scenario may not likely exist in the network, we have further discussed this scenario in Section VII.

In RL, both SARL and MARL agents require a certain degree of stability in the operating environment in order to acquire accurate learnt knowledge. In our work, a stable environment is seen as having more legitimate (or regular) SUs than malicious SUs. We consider that a SU, whether legitimate or malicious, is capable to launch attacks but with lower attack probability (and / or lower attack intensity). From the SARL clusterhead's perspective, when the attack probability (or attack intensity) increases, it causes the operating environment to become unstable, and hence it affects the outcome of the learnt knowledge about its member nodes. From the attacker's perspective, when it adopts the SARL approach, its independent learnt knowledge is more accurate as compared to adopting the MARL approach that incorporates shared knowledge from its neighboring nodes.

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

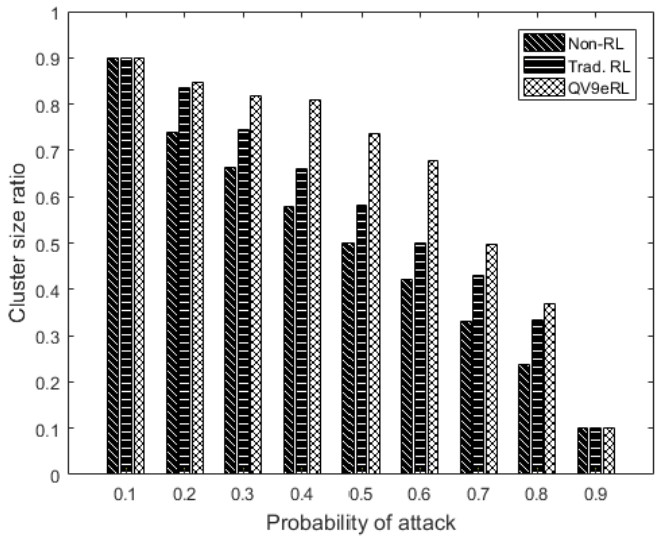

Fig. 15: Performance comparison of non-RL, traditional RL and rule-based RL algorithms — *Rule-based RL (QV9eRL) outperforms non-RL and traditional RL even when the attack probability* $\mathbb{P}_c \geq 0.7$.

As a summary, when there is a strong presence of malicious SUs, the clusterhead resources are bound to be wasted by member nodes, and hence yield lower cluster size ratio.

E. Complexity Analysis

We investigate the computational, message and storage complexities of SARL and MARL. The complexity analysis is inspired by [36] and it consists of *step-wise*, which refers to a single iteration or execution of the RL algorithm, *agent-wise* at the agent level, and *network-wide* at the network level that covers all the agents in the network.

Computational complexity defines the number of execution cycles required to update the Q-values for all state-action pairs of the agents. Using Algorithm 1, the *step-wise* complexity is $\mathcal{O}(|A|)$ since an agent i updates its Q-value upon receiving a delayed reward (Step 9) and each state has $|A|$ actions; the *agent-wise* complexity is $\mathcal{O}(|S||A|)$, since an agent i updates its Q-value for the state-action pairs, and the *network-wide* complexity is $\mathcal{O}(|I||S||A|)$ since we assume there are $|I|$ agents in the network.

Message complexity defines the number of messages being exchanged among the agents in order to update a Q-value. Using Algorithm 2, the *step-wise* complexity is $\leq |J|$ an agent i exchanges shared information with its neighbors J (Step 8) since we assume there are $|J|$ neighboring agents; the *agent-wise* complexity is $\leq |J|$, since an agent i has $|J|$ neighboring agents to exchange messages, and the *network-wide* complexity is $\leq |I||J|$ since we assume there are $|I|$ agents in the network.

Storage complexity defines the amount of memory required to store local statistics and Q-values. Using Algorithm 1, the *step-wise* complexity is 1 since an agent i stores its Q-value for a state-action pair (Step 9); the *agent-wise* complexity is $\leq (|S||A|)$ in a Q-table, since an agent i updates its Q-value for the state-action pairs, and the *network-wide* complexity is $\mathcal{O}(|I||S||A|)$ since we assume there are $|I|$ agents in the network.

VII. Conclusions and Future Work

This article presents a novel reinforcement learning (RL)-based trust model for a cluster size adjustment scheme to increase network scalability in a distributed cognitive radio network (CRN). Given the CRNs' intrinsic nature, which is dynamic in channel availability, it is necessary for the legitimate secondary users (SUs) to efficiently utilize the available white spaces in the network. However, such necessity has opened up a security challenge as some SUs may intentionally waste the white spaces causing the network to be unscalable.

The two main entities of interest are the attackers and the clusterhead. The attackers launch two main types of attack, namely non-single agent RL attack (non-SARL) (or *non-intelligence* attack), which consists of *deterministic* and *random* attacks, and RL attack (or *intelligent* attack), which consists of SARL and multi-agent RL (MARL) attacks. The simulation results have shown that the non-RL attack, namely random attack yields the same attack effects as the deterministic ($I_t^i = 1$) attack while the RL attack, namely SARL attack causes the most detrimental effects to the cluster size, followed by the MARL attack and non-SARL attack (see Fig. 10 and Fig. 13). As for the clusterhead, the proposed clusterhead rule-based SARL approach (QV9eRL) has been shown to improve cluster scalability compared to the non-SARL and traditional (SARL) approaches by increasing the cluster size ratio in the network up to 18%.

Further research can be pursued to investigate the following open issues. Firstly, multi-clusters can be established in the distributed network to facilitate token sharing amongst the clusterheads so as to best utilize the available resources in the network. For instance, when a clusterhead discovers that its member nodes are mostly malicious, it can transfer its available budget to its neighboring clusters, which may have more trustworthy member nodes. Secondly, to address the marginal improvement on the ability of the SARL clusterhead to detect malicious SUs, specifically in Scenarios (b) and (d), the SARL clusterheads in the multi-clusters scenario can adopt a flexible mode of learning, whereby the clusterheads can switch between SARL and MARL approaches under different operating environments. For instance, when the operating environment is considered stable (i.e., the probability of attack is less than 0.6 and the cluster size ratio is greater than 0.5 as shown in Figs. 9 and 10, respectively, a clusterhead chooses the MARL approach). Thirdly, from the attacker's perspective, several models such as *Strategically-Timed Attack*, which is a strategically-timed attack that selects a subset of time steps to attack, and *Enchanting Attack*, which lures the Deep RL clusterhead from its current state to a specified target state, can be investigated [37]. Finally, since MARL attackers (i.e., attackers with greater capability) do not exhibit their effectiveness of attack in a dynamic operating environment, further work can be carried out to reduce the number of collaborative

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

attackers when they launch such attack. Lesser number of MARL attackers may increase the attack effectiveness since lesser and possible inaccurate knowledge is being shared [35].

REFERENCES

[1] I.F. Akyildiz, W.Y. Lee, M.C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Elsevier Comp. Net.*, vol. 50, no. 13, pp. 2127–59, 2006.

[2] N. Dutta, H.K.D. Sarma, and Z. Polkowski, "Cluster based routing in cognitive radio adhoc networks: reconnoitering SINR and ETT impact on clustering," *Computer Communications*, vol. 115, pp. 10-20, 2018.

[3] D. Nguyen, P. Minet, T. Kunz, and L. Lamont, "New findings on the complexity of cluster head selection algorithms," *Proc. IEEE Int. Symp. on a World of Wirel., Mob. and Multimed. Netw. (WoWMoM'11)*, Lucca, Italy, 2011, pp. 1–10.

[4] Y. Saleem, K.L.A. Yau, H. Mohamad, N. Ramli, M.H. Rehmani, and Q. Ni, "Clustering and reinforcement-learning-based routing for cognitive radio networks," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 146–151, 2017.

[5] A. Ramli, and D. Grace, "Reinforcement learning-based clustering protocols for a self-organising cognitive radio network," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 4, pp. 544–556, 2016.

[6] A. Celik, and A.E. Kamal,"Multi-objective clustering optimization for multi-channel cooperative sensing in CRNs," *IEEE Trans. on Cog. Comms and Netw.*,vol. 2, no. 2, pp. 150–61, 2016.

[7] G.Y. Chang, S.Y. Wang, and Y.X. Liu, "A jamming-resistant channel hopping scheme for cognitive radio networks," *IEEE Trans. on Wireless Communications*, vol. 16, no. 10, pp. 6712–25, 2017.

[8] R.S. Sutton, and A.G. Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA, USA: MIT Press; May 1998.

[9] R. Krishnan, and D. Starobinski, "Efficient clustering algorithms for self-organizing wireless sensor networks," *Elsevier Ad Hoc Net.*, vol. 4, no. 1, pp. 36–59, 2006.

[10] F.B. Abdesslem, A. Ziviani, M.D.D. Amorim, and P. Todorova, "Looking around first: localized potential-based clustering in spontaneous networks," *IEEE Comm. Let.*, vol.11, no. 8, pp. 653–655, 2007.

[11] H. Chen, L. Xie, and X. Ni, "Reputation-based hierarchically cooperative spectrum sensing scheme in cognitive radio networks," *China communications*, vol. 11, no. 1, pp. 15–38, 2014.

[12] L. Lazos, S. Liu, and M. Krunz, "Mitigating control-channel jamming attacks in multi-channel ad hoc networks," *Proc. of second ACM Conf. on Wireless network security*, 2009, pp. 169–180.

[13] D. Das, and S. Das, "Intelligent resource allocation scheme for the cognitive radio network in the presence of primary user emulation attack," *IET Communications*, vol. 11, no. 15, pp. 2370–79, 2017.

[14] S. Bhattacharjee, S. Debroy, and M. Chatterjee,"Quantifying Trust for Robust Fusion While Spectrum Sharing in Distributed DSA Networks," *IEEE Trans. on Cog. Comms and Netw.*,vol. 3, no. 2, pp. 138–54, 2017.

[15] R. Chen, J. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," *Proc. IEEE Intl. Conf. on Computer Communications*, Phoenix, AZ, USA, April 2008, pp. 1876–1884.

[16] S. Xu, Y. Shang, and H. Wang, "Double thresholds based cooperative spectrum sensing against untrusted secondary users in cognitive radio networks," *Proc. IEEE Veh. Tech. Conf.*, Barcelona, Spain, April 2009, pp. 1–5.

[17] K. Zeng, P. Pawelczak, and D. Cabric, "Reputation-based cooperative spectrum sensing with trusted notes assistance," *IEEE Comm. Let.*, vol. 14, no. 3, pp. 226–28, 2010.

[18] P. Kaligineedi, M. Khabbazian, and V.K. Bhargava, "Secure cooperative sensing techniques for cognitive radio systems," in *Proc. IEEE Intl. Conf. Comm. (ICC)*, Beijing, China, May 2008, pp. 3406–10.

[19] H. Chen, M. Zhou, L. Xie, K. Wang, and J. Li, "Joint spectrum sensing and resource allocation scheme in cognitive radio networks with spectrum sensing data falsification attack," *IEEE Trans. Veh. Tech.*, vol. 65, no. 11, pp. 9181–91, 2016.

[20] G. Wang, and G. Cho, "Reputation-based cluster head elections in wireless sensor networks," *SAGE Publications*, vol. 89, no. 7, pp. 829–845, 2013.

[21] M.H. Ling, K.L.A. Yau, and G.S. Poh, "Trust and reputation management in cognitive radio networks: a survey," *Security Comm. Net.*, vol. 7, no. 11, pp. 2160–79, 2014.

[22] G.A Shah, F. Alagoz, E.A Fadel, and O.B. Akan, "A spectrum-aware clustering for efficient multimedia routing in cognitive radio sensor networks," *IEEE Trans. Veh. Tech.*, vol. 63, no. 7, pp. 3369–80, 2014.

[23] W. Zhang, Y. Yang, and C.K. Yeo, "Cluster-based cooperative spectrum sensing assignment strategy for heterogeneous cognitive radio network," *IEEE Trans. Veh. Tech.*, vol. 64, no. 6, pp. 2637–47, 2015.

[24] Z. Javed, K.L.A. Yau, H. Mohamad, N. Ramli, J. Qadir, and Q. Ni, "RL-Budget: A learning-based cluster size adjustment scheme for cognitive radio networks," *IEEE Access*, vol. 6, pp. 1055–72, 2018, doi: 10.1109/ACCESS.2017.2777867.

[25] F. Fu, and M.V.D. Schaar, "Learning to compete for resources in wireless stochastic games," *IEEE Trans. Veh. Tech.*, vol. 58, no. 4, pp. 1904–19, 2009.

[26] J.R. Kok, and N. Vlassis, "Collaborative multiagent reinforcement learning by payoff propagation," *Journal of Machine Learning Research*, vol. 7, pp. 1789–1828, 2006.

[27] J. Lundén, S.R. Kulkarni, V. Koivunen and H. V. Poor, "Multiagent Reinforcement Learning Based Spectrum Sensing Policies for Cognitive Radio Networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 858–868, 2013.

[28] S. Barve and P. Kulkarni,"Multi-Agent Reinforcement Learning Based Opportunistic Routing and Channel Assignment for Mobile Cognitive Radio Ad Hoc Network," *Mobile Networks and Applications*, vol. 19, no. 6, pp. 720–730, 2014.

[29] S. El-Tantawy, B. Abdulhai and H. Abdelgawad, "Multiagent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers (MARLIN-ATSC): Methodology and Large-Scale Application on Downtown Toronto," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1140–1150, 2013.

[30] M. Maskery, V. Krishnamurthy and Q. Zhao, " Decentralized dynamic spectrum access for cognitive radios: cooperative design of a non-cooperative game," *IEEE Transactions on Communications*, vol. 57, no. 2, pp. 459-469, 2009.

[31] A. Garro, "Computing Nash Equilibria in Non-Cooperative Games: An Agent-Based approach," *International Journal of Intelligent Mechatronics and Robotics*, vol.3, no. 3, pp 29–42, 2013.

[32] E. Rasmusen, B. Blackwell, "Games and information," *Cambridge, MA*, vol. 15, 1994.

[33] M.S. Stankovic, and S.S. Stankovic,"Multi-agent temporal-difference learning with linear function approximation: Weak convergence under time-varying network topologies,"*American Control Conference (ACC)*, Boston, MA, USA, August 2016, Electronic ISSN: 2378-5861, doi: 10.1109/ACC.2016.7524910.

[34] K. Tuyls, and G. Weiss,"Multiagent learning: basics, challenges, and prospects," *AI Magazine*, vol. 33, no. 3, pp. 41–52, 2012.

[35] L. Busoniu, R. Babuska, and B.D. Schutter,"A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, vol. 38, no. 2, pp. 156–172, 2008.

[36] C. Bettstetter, and S. Konig, "On the message and time of a distributed mobility-adaptive clustering algorithm in wireless ad hoc networks," in *Proceedings of the 4th European Wireless Conference. (EW'02)*, Florence, Italy, 2002, 128–134.

[37] Y.C. Lin, Z.W. Hong,Y.H. Liao, M.L. Shih, M.Y. Liu, M. Sun, "Tactics of adversarial attack on deep reinforcement learning agents," in *arXiv preprint arXiv:1703.06748, 2017*, 2017.

Mee Hong Ling has a Bachelor (Hons.) degree in Computer and Mathematical studies from Oxford Brookes University, UK and a Master degree in Data Engineering (Computer Science) from Keele University, UK. She lectures in the Department of Computing and Information Systems at Sunway University. She is currently pursuing her PhD degree in Computing. Her research interests are in the areas of security, cognitive radio networks and applied reinforcement learning.

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

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/TCCN.2018.2881135, IEEE Transactions on Cognitive Communications and Networking

Kok-Lim Alvin Yau received his B.Eng. degree in electrical and electronics engineering (First Class Honors) from the Universiti Teknologi Petronas, Malaysia (2005), his M.Sc. (electrical engineering) from the National University of Singapore (2007), and his Ph.D. (network engineering) from Victoria University of Wellington, New Zealand (2010). He was awarded the 2007 Professional Engineer Board of Singapore Gold Medal for being the best graduate of the M.Sc. degree in 2006/07. Currently, he is an associate professor at Sunway University, Malaysia. He researches, lectures, and consults in cognitive radio, wireless networking, and applied artificial intelligence.

Junaid Qadir Junaid Qadir is currently an Associate Professor at the Information Technology University (ITU) Punjab in Lahore, Pakistan. He completed his B.S. in Electrical Engineering from UET, Lahore, Pakistan and his Ph.D. from University of New South Wales, Australia in 2008. He served as an Assistant Professor at the School of Electrical Engineering and Computer Sciences (SEECS), National University of Sciences and Technology (NUST), Pakistan from 2008 to 2015 where he also directed the Cognet Lab at SEECS. He has been awarded the highest national teaching award in Pakistan—the higher education commission's (HEC) best university teacher award—for the year 2012-2013. He has been nominated for this award twice (2011, and 2012-2013). His research interests include the application of algorithmic, machine learning, and optimization techniques in networks. In particular, he is interested in the broad areas of wireless networks, cognitive networking, software-defined networks, and cloud computing. He is a regular reviewer for a number of journals and has served in the program committee of a number of international conferences. He serves as an Associate Editor for IEEE Access, IEEE Communications Magazine, and Big Data Analytics. He is a member of ACM, and a senior member of IEEE.

Qiang Ni received his B.Sc., M.Sc., and Ph.D. degrees in engineering from Huazhong University of Science and Technology, Wuhan, China, in 1993, 1996, and 1999, respectively. He is a professor of communications and networking with the School of Computing and Communications, Lancaster University, United Kingdom. He previously led the Intelligent Wireless Communication Networking Group at Brunel University London. His main research interests include wireless communications and networking. He has published more than 120 papers in his areas of interest. He was an IEEE 802.11 Wireless Standard Working Group voting member and a contributor to the IEEE wireless standards.

2332-7731 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.