This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2022.3184112

Strategic Honeypot Deployment in Ultra-Dense Beyond 5G Networks: A Reinforcement Learning Approach

Panagiotis Radoglou-Grammatikis[†], Panagiotis Sarigiannidis[†], Panagiotis Diamantoulakis[‡], Thomas Lagkas[§], Theocharis Saoulidis[¶], Eleftherios Fountoukidis[‖] and George Karagiannidis[‡]

***Abstract*—The progression of Software Defined Networking (SDN) and the virtualisation technologies lead to the beyond 5G era, providing multiple benefits in the smart economies. However, despite the advantages, security issues still remain. In particular, SDN/NFV and cloud/edge computing are related to various security issues. Moreover, due to the wireless nature of the entities, they are prone to a wide range of cyberthreats. Therefore, the presence of appropriate intrusion detection mechanisms is critical. Although both Machine Learning (ML) and Deep Learning (DL) have optimised the typical rule-based detection systems, the use of ML and DL requires labelled pre-existing datasets. However, this kind of data varies based on the nature of the respective environment. Another smart solution for detecting intrusions is to use honeypots. A honeypot acts as a decoy with the goal to mislead the cyberatatcker and protect the real assets. In this paper, we focus on Wireless Honeypots (WHs) in ultra-dense networks. In particular, we introduce a strategic honeypot deployment method, using two Reinforcement Learning (RL) techniques: (a) $e-Greedy$ and (b) $Q-Learning$. Both methods aim to identify the optimal number of honeypots that can be deployed for protecting the actual entities. The experimental results demonstrate the efficacy of both methods.**

***Index Terms*—Honeypot, Intrusion Detection, Reinforcement Learning, Wireless Communication**

I. Introduction

Through the evolution of the softwarisation and virtualisation technologies, such as Software Defined Networking (SDN), Network Function Virtualisation (NFV) and cloud/edge computing, 5G has become a digital reality, providing multiple benefits to the individuals' aspects, such as higher connectivity, lower latency and improved energy efficiency. Already, most of the developed countries offer commercial 5G services. Based on the 5G Public-Private Partnership (5G-PPP), 5G will be able to connect approximately seven trillion wireless entities [1]. Therefore, many Internet of Things (IoT) and Industrial IoT (IIoT) applications such as the smart electrical grid and remote healthcare services, will benefit from 5G. However, the aforementioned technologies are characterised by several security issues [2], [3]. In [1], A. Ahmad et al. provide a detailed overview about the 5G security challenges. Other similar studies are listed in [4], [5]. Moreover, it is noteworthy that despite the security characteristics of 5G, such as the sufficient encryption mechanisms, the wireless systems within the Radio Access network (RAN) are prone also to various cyberthreats from their first-generation (1G). Their evolution even beyond 5G (B5G) or 6G can lead to new sophisticated and complicated cyberattacks with devastating consequences.

Based on the aforementioned remarks, it is evident that the presence of efficient intrusion detection mechanisms is necessary. The rise of Artificial Intelligence (AI) techniques, such as Machine Learning (ML) and Deep Learning (DL), has evolved significantly the conventional signature and specification-based Intrusion Detection Systems (IDS). Many studies investigate in detail the efficiency of ML and DL-based IDS [6], [7]. In particular, through ML and DL, the current IDS are capable of detecting and discriminating unknown anomalies and zero-day cyberattacks. However, in contrast to signature/specification-based IDS, ML and DL-based IDS are usually linked to a high number of misclassifications due to the presence of False Positive (FP) and False Negative (FN) results. Moreover, ML and DL require the existence of a labelled dataset that can differ from environment to environment. Due to the sensitive nature of this kind of data, usually, there are not publicly available intrusion detection datasets especially related to the 5G domain. Another smart detection mechanism that can contribute to the timely detection of a cyberattacker is a honeypot. A honeypot is an intentional security hole that aims to mislead the cyberattackers and protect the real assets. However, it is noteworthy that despite the defensive nature of the honeypot, it can also be used by a cyberattacker to reach the real assets.

The goal of this paper is twofold. First, we focus our attention on deploying honeypots in a strategic manner, taking full advantage of Reinforcement Learning (RL). In particular, the deployment problem is transformed into a Multi-Armed Bandit Problem (MAB), where our goal is to deploy the optimal number of honeypots, taking into account the benefits and costs of the *Defender* and the *Attacker*. In particular, we

*This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833955.

† P. Radoglou Grammatikis and P. Sarigiannidis are with the Department of Electrical and Computer Engineering, University of Western Macedonia, Kozani 50100, Greece - E-Mail: {pradoglou, psarigiannidis}@uowm.gr

‡ P. Diamantoulakis and G. Karagiannidis are with Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece (E-Mails: {padiaman,geokarag}@auth.gr)

§ T. Lagkas is with the Department of Computer Science, International Hellenic University, Kavala Campus, 65404, Greece - E-Mail: tlagkas@cs.ihu.gr

¶ T. Saoulidis and E. Fountoukidis is with Sidroco Holdings Ltd, Petraki Giallourou 22, Office 11, 1077 Nicosia, Cyprus - E-Mail: asarigia@sidroco.com

adopt two RL methods: (a) $e-Greedy$ and (b) $Q-Learning$. Based on the various security events detected by the honeypots or other detection systems, both $e-Greedy$ and $Q-Learning$ converge to the appropriate number of honeypots that should be deployed. Second, we first introduce the theoretic framework of the presence and the role of Wireless Honeypots (WHs) in ultra-dense networks. Finally, we evaluate the RL-based honeypot deployment methods with respect to deploying WHs in ultra-dense networks. Consequently, the contribution of this work is summarised in the following bullet points.

- **Strategic RL-based Honeypot Deployment**: We introduce an RL-based honeypot deployment method, taking advantage of $e-Greedy$ and $Q-Learning$. The proposed method identifies how many honeypots can be deployed in an infrastructure, taking into various costs and benefits of the *Defender* and the *Attacker*.
- **Wireless Honeypots in Ultra-Dense Networks**: We first introduce the role and use of WHs in ultra-dense networks in order to mitigate security risks. For this purpose, the impact of density in wireless networks is investigated and modelled. Finally, we evaluate the above RL-based honeypot deployment method in ultra-dense networks.

The rest of this paper is organised as follows. Section II provides a background about honeypots and RL. Section III presents some relevant works and discusses our contribution. In section IV, the concept of the honeypot orchestrator is provided as an RL agent. In section V, we introduce the strategic RL-based honeypot deployment method. Section VI investigates the deployment of WHs in ultra-dense networks. Next, section VII focuses on the evaluation analysis with respect to deploying WHs in ultra-dense networks. Finally, section VIII concludes this work.

II. Background

A. Honeypot: A Security Trap

Honeypots are assets with no production value that imitate the behaviour of the real assets, thereby protecting them and collecting valuable information about the cyberattackers. In particular, the honeypots can be classified into two categories: (a) production honeypots and (b) research honeypots. The production honeypots are placed into the production network, trying to hide the real assets from potential malicious insiders. On the other side, the research honeypots are exposed to public networks like the Internet, attracting potential cyber-attackers and collecting important information related to their behaviour. It is noteworthy that any interaction with a honeypot is considered suspicious since the legitimate users do not have any reason to interact with it. Moreover, the honeypots can be classified based on the interaction level as (a) Low-Interaction Honeypots (LIH), (b) Medium-Interaction Honeypots (MIH) and (c) High-Interaction Honeypots (HIH). LIH can simulate some network services in terms of the various communication protocols, without emulating completely the network behaviour of the real assets. MIH can emulate better the network behaviour of the real assets, transmitting, for instance, similar network packets as the real entity. Finally, HIP represents a complete copy of the real asset, comprising all of its hardware and software characteristics.

Both academia and industry have implemented several honeypots. In particular, *Deception Toolkit (DTK)* [8] was the first honeypot released in 1997, emulating known vulnerabilities of UNIX. *HoneyBOT* [9] is a LIH for Windows Operating Systems (OS), simulating relevant vulnerabilities. Similarly, *KFSensor* [10] is a commercial honeypot for Windows OS. *HoneyD* [11] is probably the most known honeypot capable of emulating at the same time multiple hosts. *Tiny Honeypot* [12] is a server-based honeypot, which listens to all Transmission Control Protocol (TCP) ports, logging all interaction activities. *Dionaea* [13] is written in Python and emulates the MQ Telemetry Transport (MQTT) protocol. *Jackpot* [12] is related to Simple Mail Transfer Protocol (SMTP) and aims to combat email spam. *Cowrie* [14] is a LIH emulating SSH. *Conpot* [15] is an industrial honeypot emulating multiple relevant protocols like Modbus and IEC 60870-5-104. In addition, an overview of WHs along history is discussed in [16], where they are defined as nodes that offer wireless access whose value is being probed, attacked, or compromised, letting the attackers to interact with them. In more detail, the main goal of WHs is to gather information about the attacks performed on wireless networks and the associated technologies, focusing on the attacks that exploit the wireless technologies' weaknesses, which are mainly due to the use of unguided transmission medium [17]. The main principles of the WHs can be used in several types of networks, including cellular, Local Area Networks (LANs), sensor networks and Unmanned Aerial Vehicles (UAVs)-based networks [18].

Many supporting tools have been developed in order to analyse the data retrieved from honeypots or to extend their functionalities [19]. In particular, *Bait-n-Switch* [20] aims to redirect all malicious traffic to a honeypot. Accordingly, *Honeynet Security Console (HSC)* [21] analyses, correlates and visualises honeypots logs. *Honeysnap* [22] processes Packet Capture (PCAP) files that were collected by server-based honeypots. *GSOC-Honeyweb* [12] is devoted to the management of client-based honeypots via a user-friendly environment. Moreover, *TraCINg* [12] aggregates data from multiple honeypots and correlates this information in order to discover possible worms.

It is noteworthy that many honeypots projects have been organised in order to exploit at the maximum level the benefits of honeypots and discover potential zero-day attacks. In particular, the *Honeynet Project* was started in 1999 to explore and investigate zero-day cyberattacks. Furthermore, the *Leurre.com* project [23] deployed multiple LIHs in more than 30 counties, aiming at collecting quantitative data related to cyberthreats and vulnerabilities. Accordingly, *NoAH-Project* coordinated by Foundation for Research and Technology Hellas (FORTH) deployed an HIH called Argos [24] to enhance the protection of Internet Service Providers (ISPs) and investigate potential zero-day attacks. The *mw-collect Alliance* project collected information about various malware by deploying multiple Nepenthes sensors [12]. Moreover, *Telekom-Fruhwarnsystem* [12] was started in 2013 to collect various datasets related to honeypot activities. Finally, *H2020*

SPEAR [25] and *H2020 SDN-microSENSE* [26] implemented various industrial honeypots for the smart electrical grid.

B. Reinforcement Learning

The goal of an RL agent is to identify the optimal policy performed in an environment based on the various states and the possible actions. An action a_t can be performed at time t in the state (s_t), thus leading to a new state s_{t+1} and a reward $R(s_t, a_t)$. The optimal policy refers to maximising the accumulated rewards over time. There are various kinds of RL methods, such as $e-$*Greedy*, *Thompson Sampling*, *SARSA*, $Q-$*Learning* and *Deep* $Q-$*Learning*. In this paper, we focus on $e-$*Greedy* and $Q-$*Learning*. However, after defining the environment with respect to the available states and actions, each of the aforementioned methods follows two phases: (a) training process and (b) inference. During the training process, the RL model (based on the corresponding method) is trained to identify the best policy. In particular, after initialising the parameters of each method, the RL model starts interacting with the environment, thus leading to new states and obtaining the corresponding rewards. At the end of each episode, the parameters are adjusted appropriately in order to gain a better reward during the next episode. This process is repeated till convergence. Finally, inference follows, which means that the RL model is ready to be used in the environment without adjusting the parameters of the RL method. More information about the various RL methods is given in [27].

III. Related Work

Several studies have investigated the role of honeypots and relevant optimisation techniques with AI and game theory in order to protect critical organisations and infrastructures. Some of them are listed below [12], [28]–[34]. In particular, in [28], J. Franco et al. provide a survey about honeypots and honeynets for the IoT and IIoT. In [12], M. Nawrocki et al. present a comprehensive study about honeypot software and relevant data analytics. Similarly, in [29], the authors discuss the decoy and security operations of honeypots, presenting a detailed taxonomy. On the other hand, in [30], C. Dalamgkas et al. focus on honeypots related to the smart electrical grid. In [31], C. Kiekintveld et al. present a study about game theory methods used to deploy honeypots in an efficient manner, modelling the behaviour of the attacker and the defender. In [32], L. Shi et al. investigate the performance of honeypots through Petri nets. In [33], W. Zhang et al. present a honeynet composed of multiport honeypots for countering IoT attacks. Finally, in [34], L. Shi et al. provide a blockchain-based dynamic and distributed honeypot. Next, we discuss some relevant works in a more detailed manner and show the novel points of our paper. Each paragraph focuses on a separate paper.

In [35], P. Radoglou-Grammatikis et al. provide TRUSTY. TRUSTY is a web-based platform capable of collecting, normalising and processing security logs originating from honeypot applications. In particular, the authors focused mainly on industrial honeypots, thus using TRUSTY to generate a dataset related to honeypot events. Based on this dataset, a strategic method for deploying honeypots in a smart electrical grid environment is also provided. First, the behaviour of the attacker and the defender is modelled in terms of the various costs and benefits with respect to attacking a real asset or a honeypot. Consequently, the utility functions of the attacker and the defender are defined, respectively. Next, the deployment process is formed as a Multi-Armed Bandit (MAB) problem with the goal to optimise the utility function of the defender. The MAB problem is solved through the $e-Greedy$ method. The evaluation results demonstrate the efficiency of the proposed deployment method with respect to selecting the optimal number of honeypots.

In [36], P. Diamantoulakis et al. present a sophisticated honeypot deployment method, taking full advantage of game theory. After defining the utility function of the defender and the attacker, a one-shot game is formulated. For this purpose, the various costs and benefits for the defender and the attacker are determined, respectively. Next, the solution of this game is given by calculating the Nash Equilibrium (NE). If NE is not available, the decision of the defender is modelled through a non-convex min-max optimisation analysis. Subsequently, the authors investigate a continuous scenario related to the previous one-shot game. This means that the defender and the attacker play the one-shot game more than one time. Thus, a Bayesian game is modelled, and the corresponding Bayesian NE (BNE) is determined. The simulation results demonstrate the effectiveness of each method regarding the selection of the optimal number of honeypots in a smart electrical grid environment.

In [37], K. Wang et al. introduce a Bayesian honeypot model in order to protect an Advanced Metering Infrastructure (AMI) against Denial of Service (DoS) attacks. In particular, the authors investigate three cases provided by a service provider: (a) a real AMI communication, (b) honeypot service and (c) anti-honeypot service. The first two are related to a legitimate user, while the anti-honeypot services refer to actions performed by a cyberattacker in order to recognise the presence of honeypots and bypass them. The goal is to balance the detection rate and the energy consumption. Thus optimal strategies are defined for the attacker and the defender. Next, several BNEs are identified. The experimental results show that the proposed game can enhance the honeypots' detection rate and the energy consumption.

In [38], Y. Zhang et al. introduce an adaptive honeypot deployment mechanism based on Learning Automata (LA). LA is an RL method used to select an optimal action based on a finite set of actions and the interactions with a random environment. LA can be defined as a tuple of five elements, namely (a) actions, (b) rewards, (c) states, (d) state transfer function and (e) output function. An attack-defence scenario is formed with two players (a) attacker and (b) defender. The actions of the attacker fall into two main phases: (a) the preparation phase and (b) the attack phase. The first one refers to the preparation activities before the execution of the attack, while the attack phase denotes the actual malicious activities. On the other hand, the actions of the defender can also be classified into two main phases: (a) the planning phase and (b) the defending phase. The planning phase indicates the

deployment of the necessary defensive mechanisms, while the defence phase refers to the countermeasures applied during the execution of a cyberattack. The proposed method considers the entirety of the nodes as the LA, and each node is considered as an action. Based on the malicious activities and the evolution of the LA, a particular number of honeypots are deployed. The experimental results show the efficacy of the proposed method in terms of the honeypots' detection rate and selecting the appropriate number of honeypots.

In [39], M. Du and K. Wang investigate the role of honeypots against Distributed DoS (DDoS) in SDN environments. First, the authors provide an anti-honeypot strategy capable of identifying the presence of honeypots in an SDN network. In particular, the first step of the anti-honeypot strategy is to identify whether there is a honeypot in the SDN network. Next, the honeypot type is clarified, and the optimal attack strategy is determined. To protect the SDN network from the above anti-honeypot strategy, the authors provide also a Bayesian pseudo-honeypot game with respect to the deployment of various kinds of honeypots in an SDN network. The authors also show the existence of several BNEs and prove that the proposed BNEs can accomplish the optimal equilibrium between the legitimate users and attackers. The evaluation results demonstrate that the proposed method can effectively counter DDoS attacks with low energy consumption.

In [40], U. Bartwal et al. provide a Security Orchestration Automation and Response (SOAR) engine that deploys honeypots based on security events related to DDoS and botnets. In particular, the proposed SOAR engine is composed of ten architectural components: (a) Host Machine, (b) Virtual Machines, (c) Honeypots, (d) Container Registry, (e) Storage, (f) Traffic Tracker, (g) Botnet Detector, (h) DDoS Detector, (i) Orchestration Engine and (j) Access Logs. The orchestration engine is responsible for deploying the honeypots located in the Container Registry based on the security events recognised by the Botnet and DDoS detectors. The detectors adopt both Machine Learning (ML) and signature/specification rules. Initially, no honeypot is deployed. Next, based on the security events, the orchestration engine undertakes to start the first honeypots. If the attackers start interacting with the honeypots, new honeypots are deployed by the orchestration engine, thus minimising the attack probability against the real assets.

In [41], W. Fan et al. present HoneyDoc, an SDN-based architecture about the honeypot deployment. The architectural model of SDN consists of three main planes: (a) Data Plane, (b) Control Plane and (c) Application Plane. The data plane refers to the physical and virtualised entities connected to SDN switches. Next, the control plane is devoted to the SDN controllers responsible for orchestrating and managing the SDN switches. Finally, the application plane refers to the SDN application that can interact with the SDN controller. HoneyDoc is composed of three main modules: (a) Decoy Manager, (b) Captor Manager and (c) Orchestration Core. The Decoy Manager is responsible for deploying the various honeypots, including LIH, MIH and HIH. All the honeypots are located in the control plane. Next, the Capture Manager refers to an SDN application consisting of three submodules, namely (a) Data Capture, (b) Data Control and (c) Data Analysis, responsible for capturing, controlling and analysing the honeypot data, respectively. Finally, the Orchestration Core is located in the Control Plane and is responsible for coordinating the actions of the Decoy and Captor Managers.

Undoubtedly, the previous works provide useful insights, methodologies and tools. Several papers adopt game theory and RL methods in order to deploy honeypots in a strategic manner. Characteristic examples are [36], [37], [39]. However, despite the evaluation results, this kind of modelling cannot be adopted easily during the production mode of real environments. Moreover, the parameters of the game models should be re-adjusted based on the impact of the various security events and alarms. On the other hand, the previous RL methods do not consider the detection of security events through other detection mechanisms than honeypots. Finally, it is worth noting that the current works do not consider the use of WHs in ultra-dense networks. Based on the aforementioned remarks, in this paper, we introduce first an RL-based honeypot deployment method modelling the behaviour of the *Defender* with the use of WHs and other detection measures. In terms of 5G, a WH can emulate a vulnerable gNB. The smart deployment of the WHs is modelled as security game in terms of the costs and benefits of the *Defender*. The security game is solved through two RL methods, namely: (a) $e - Greedy$ and (b) $Q - Learning$. Finally, we model and introduce in a theoretic manner the use of WHs in ultra-dense networking environments.

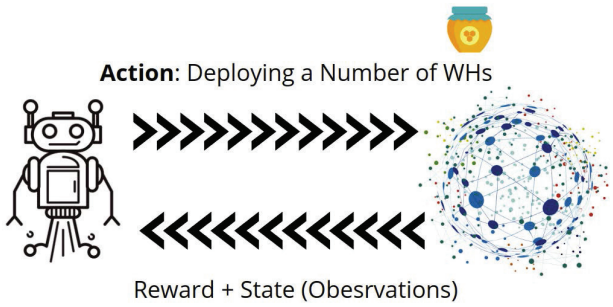

Fig. 1. RL-based Security Game: Deploying a Number of Honeypots in Ultra-Dense Networks

IV. Security Analysis

Based on the aforementioned remarks, Fig. 1 illustrates the goal of our RL-based security agent in the context of an ultra-dense networking environment. Based on the various security events, the RL agents tries to deploy the appropriate number of WHs in order to protect the real access points. In the context of 5G networks, the access points can refer to gNB. Therefore, the RL agent play the role of a honeypot orchestrator, deploying the appropriate number of honeypots. To this end, the unique characteristics of the ultra-dense network should be considered. Thus, the utility function of the *Defender* should take into account not only the security characteristics but also the quality of the network in terms of the services provided. Thus, the RL agent interacts with the environment and receives an corresponding reward and state (i.e., ste of observations about the security and the quality of

the network) given the new security events. For each security events detected either by WHs or other detection mechanisms, the honeypot orchestrator is triggered with respect to deploying the appropriate number of honeypots.

TABLE I
SYMBOLS AND NOTATION

Notation	Explanation
N	The number of the real access points and honeypots that are connected.
$U_D[t]$	The utility of the *Defender* at the time interval t.
$g()$	A function expressing initially the utility of the *Defender* at the time interval t.
$S_{D,i}$	The strategy of the *Defender* regarding each asset.
$S_{A,i}$	The strategy of the *Attacker* regarding each asset.
$I_{r,i}$	It denotes whether the attack against the asset i is detected.
δ_1	The benefit of the *Defender* for each attack against a honeypot.
δ_2	The benefit of the *Defender* for each attack against an access point is detected.
δ_3	The cost of the *Defender* for each attack not detected.
θ	The probability that an asset is used as a honeypot.
ϕ	The probability that a real asset is under an attack.
$\tilde{U}_D$	The expected value of the *Defender's* utility.
P_r	The probability that an attack against a real asset is detected.
C	The cost induced by the use of honeypots.
$\tilde{C}$	The expected value of the cost induced by the use of honeypots.
λ	The density of the remote radio heads deployment.
L	The number of the deployed access points.
M	The number of the deployed WHs.
d_i	The distance between the i-th user and its closest AP.
$f_{X,i}(x)$	The probability density function of x_i.
$F_{X,i}(x)$	The cumulative density function of x_i.
R_i	The achievable communication rate of user i.
γ_i	The signal-to-noise ratio at the reference distance of 1 m.
h_i	The small scale fading power gain of user i.
β	The path loss exponent.
P_i	The transmit power of user i.
σ^2	The power of the additive white Gaussian noise.
L_{ref}	The path loss at the reference distance.
$R_{t,i}$	The transmission rate of the i-th user.
$P_{i,out}$	The outage probability of the i-th user.
Z	The number of re-transmissions.
$\mathcal{P}(\cdot)$	It denotes probability.
$\exp(\cdot)$	It denotes the exponential function.
κ	The parameter of the exponential distribution.
C_i	The cost induced to the i-th user due to the use of honeypots .
$\mathbb{E}[\cdot]$	It denotes expectation.
p_{uc}	The price in the unit commitment stage.
p_{ed}	The price in the economic-dispatch stage.
μ_i	The mean energy demand of the i-th device.
E_{max}	The maximum energy consumption of the i-th device.
r	The actual energy consumption of the i-th device.
S	The space of states
A	The space of actions
s_t	The current state at time t
a_t	The action performed in the state s_t
$R(s_t, a_t)$	The reward of action a_t in the state s_t
TD	Temporal Difference
SE	A set of security events
$a_{LearningRate}$	The learning rate, which denotes how fast the Q values are updated

V. STRATEGIC HONEYPOT DEPLOYMENT WITH REINFORCEMENT LEARNING

We consider the honeypot deployment problem as a security game with two antagonistic players: (a) *Attacker* and (b) *Defender*. The goal of the *Attacker* is to attack the real access points, while the *Defender* intends to deploy/use the appropriate number of honeypots that will provide the maximum protection, taking into account the available computing resources and the behaviour of the *Attacker*. Let N be the total number of the connected stations that can serve either as honeypots or access points. The ratio of N utilised by honeypots is symbolised by θ. $s_{D,i} \in \{-1, 1\}$ represents the strategy of the *Defender*. $s_{D,i}$ equals -1 and 1 when the cyberattack targets a real access point or a honeypot, respectively. Similarly, δ_1 defines the benefit related to the *Defender* for each attack against a honeypot, while δ_2 implies the benefit of the *Defender* for each attack detected without the use of a honeypot. Finally, δ_3 is the cost of the *Defender* for each attack not detected in a timely manner. For the sake of clarity, Table I summarises the notation.

The utility function of the *Defender* in a time interval t i.e., $U_D[t]$, is given by Equation 1.

$$U_D[t] = g(\sum_{i=1}^{N} \frac{1 - S_{D,i}}{2} s_{A,i}, \sum_{i=1}^{N} \frac{1 + S_{D,i}}{2} s_{A,i}, \sum_{i=1}^{N} \frac{1 + s_{D,i}}{2}, I_{r,i}). \tag{1}$$

In Equation 1, $I_{r,i}$ is equal to 1, when the attack is detected by the node i and equal to 0 when it is not detected. Of course, when $S_{D,i} = -1$, i.e., the attacked device is a honeypot, $I_{r,i} = 1$, while if $S_{D,i}$, $I_{r,i} \in \{0, 1\}$ is a random variable. Also, $g(\cdot)$ increases in terms of $\sum_{i=1}^{N} \frac{1-S_{D,i}}{2} s_{A,i}$ and $\sum_{i=1}^{N} \frac{1+S_{D,i}}{2} s_{A,i} I_{r,i}$ and decreases in terms of $\sum_{i=1}^{N} \frac{1+S_{D,i}}{2} s_{A,i}(1 - I_{r,i})$, and $\sum_{i=1}^{N} \frac{1+s_{D,i}}{2}$. If we assume that the terms of Equation 1 progress linearly, the Equation 1 can be written in the form of Equation 2, where C is related to the cost induced by the use of honeypots (e.g., due to the use of extra resources or due to the degradation of the system's performance).

$$U_D[t] = \delta_1 \sum_{i=1}^{N} \frac{1 - S_{D,i}}{2} s_{A,i} + \delta_2 \sum_{i=1}^{N} \frac{1 + S_{D,i}}{2} s_{A,i} I_{r,i} - \delta_3 \sum_{i=1}^{N} \frac{1 + S_{D,i}}{2} s_{A,i}(1 - I_{r,i}) - C \left(\sum_{i=1}^{N} \frac{1 + s_{D,i}}{2} \right), \tag{2}$$

The best strategy for the *Defender* is to randomly allocate the honeypots so that the *Attacker* will not be able to recognise their presence. Since the *Defender* cannot know a priori the number of attacks, the goal is to optimise the expected value of U_D, denoted by $\tilde{U}_D$. This can be achieved by knowing the probability ϕ that each connected device receives an attack and by controlling the probability related to the portion of the assets that correspond to honeypots, i.e., θ. Thus, the expected

value of the *Defender's* utility function can be written by Equation 3.

$$\tilde{U}_D = \delta_1\theta\phi N + \delta_2(1-\theta)\phi N P_r - \delta_3(1-\theta)\phi(1-P_r)N - \tilde{C}(\theta) \quad (3)$$

Also, P_r is the probability that an attack is detected without the presence of a honeypot. It is worth noting that in the case where $P_r = 1$, the use of honeypots does not offer any gain to the *Defender*. Moreover, $\tilde{g}(\theta)$ is related to the expected cost induced by the use of honeypots. Therefore, based on the security events detected by the honeypots and other potential detection mechanisms, such as signature-based detection systems and ML/DL-based classification, our goal is to define the appropriate θ in order to maximise $U_D[t]$ (Equation 3). To re-define, the appropriate value of θ for each security event in the time interval t can be expressed as a MAB problem, where exploitation intends to maximise $U_D[t]$ (Equation 3) and exploration aims to check different values of θ to discover more information for the *Attacker*. In particular, the deployment process plays the role of the gambler and the various values of θ represent the slot machines. To solve the MAB problem, we adopt first the $e - Greedy$ method (Algorithm 1), where we commonly select that mean of θ providing the maximum value $\tilde{U}_D[t]$ and there is a small probability e where other values of θ are selected in order to discover how Equation 2 ranges. However, although $e-$*Greedy* is a suitable option about the exploration, sometimes, we choose a sub-optimal action randomly. Thus, we also use $Q -$ *Learning* (Algorithm 2) in order to avoid this situation. In both algorithms. data $Data$ denotes the input data, while $Result$ indicates the output of the algorithm. The number of honeypots already deployed denotes the current state s and the number of honeypots that can be deployed in a subsequent security event represents the possible actions a. In a specific case, all the states are defined in the space S, while all the actions are defined in the space A. Both S and A rely on N. Finally, the reward $R(a_t, s_t)$ of each action a_t performed in the state s_t is given by Equation 2. The functionality of $Q -$ *Learning* relies on (a) the $Q(s, a)$ values, (b) Temporal Difference $TD_t(s_t, a_t)$ (Equation 4) and (c) the Bellman equation (Equation 5). $Q(s, a)$ represents the estimated reward of the action a performed in the state s. Next, $TD_t(s_t, a_t)$ expresses the difference between $R(s_t, a_t)+\gamma max_a(Q(s_{t+1}, a))$ and $Q(s_t, a_t)$. $R(s_t, a_t) + \gamma max_a(Q(s_{t+1}, a))$ denotes the reward $R(s_t, a_t)$ received by executing the action a_t in the state s_t plus the Q value of the most optimal action executed in the future state s_{t+1} discounted by a factor $\gamma \in [0, 1]$. During the training process, by interacting with the environment, $Q -$ *Learning* intends to identify a high reward $R(s_t, a_t)$ and increase the respective $Q(s_t, a_t)$. At some point in the course of the training process, $Q -$ *Learning* will identify all the transitions leading to high rewards and high Q values. At this point, TD will decrease. In order to update the Q values for each security event, the Bellman equation is used. For each new security event detected, the Q values are updated from $t - 1$ (i.e., when the previous security event received) to t (i.e., the current security event). $a_{\text{LearningRate}} \in [0, 1]$ represents the learning rate, which denotes how fast the Q values are updated. $Q -$ *Learning* is an off-policy method. This means that the actions can be dictated by an action selection policy (i.e., behaviour policy), such as $e -$ *Greedy*, however, with respect to the training procedure, always the greedy option (i.e., target policy) is chosen.

$$TD_t(s_t, a_t) = R(s_t, a_t) + \gamma max_a(Q(s_{t+1}, a)) - Q(s_t, a_t) \quad (4)$$

$$Q_t(s_t, a_t) = Q_{t-1}(s_t, a_t) + a_{LearningRate} TD_t(s_t, a_t) \quad (5)$$

Algorithm 1: e-Greedy Honeypot Deployment

Data: N_{max}, N, UD_Matrix, sum_θ_Matrix, $mean\_\theta\_Matrix$, max_mean, securityEventCounter, δ_1, δ_2, δ_3, $\tilde{C}(\theta)$, e
Result: $a_{selected} = \theta_{selected}$
$size\_\theta\_Matrix = []$, $UD\_Matrix = []$, $sum\_\theta\_Matrix = []$, $mean\_\theta\_Matrix = []$, securityEventCounter = 0, max_mean = 0, $a = \theta_{selected}$ = 0, δ_1, δ_2, δ_3, $\tilde{C}(\theta)$ = init(), $e = 0.1$;
while *True* **do**
  Receive a security event;
  securityEventCounter = securityEventCounter +1;
  max_mean = 0;
  p = random number in [0,1];
  if $p < e$ **then**
    $\theta_{selected}$ = random integer number in [1, N];
    $U_D[t] = \delta_1 \sum_{i=1}^{N} \frac{1-S_{D,i}}{2} s_{A,i} + \delta_2 \sum_{i=1}^{N} \frac{1+S_{D,i}}{2} s_{A,i} I_{r,i} - \delta_3 \sum_{i=1}^{N} \frac{1+S_{D,i}}{2} s_{A,i}(1 - I_{r,i}) - C(\sum_{i=1}^{N} \frac{1+s_{D,i}}{2})$
    $sum\_\theta\_Matrix[\theta]$ = $sum\_\theta\_Matrix[\theta]$ + $UD\_Matrix[\theta]$;
    $mean\_\theta\_Matrix$ = $sum\_\theta\_Matrix[\theta]$ / securityEventCounter;
  end
  else
    for $\theta \leftarrow 1$ **to** N **by** 1 **do**
      $U_D[t] = \delta_1 \sum_{i=1}^{N} \frac{1-S_{D,i}}{2} s_{A,i} + \delta_2 \sum_{i=1}^{N} \frac{1+S_{D,i}}{2} s_{A,i} I_{r,i} - \delta_3 \sum_{i=1}^{N} \frac{1+S_{D,i}}{2} s_{A,i}(1 - I_{r,i}) - C(\sum_{i=1}^{N} \frac{1+s_{D,i}}{2})$
      $sum\_\theta\_Matrix[\theta]$ = $sum\_\theta\_Matrix[\theta]$ + $UD\_Matrix[\theta]$;
      $mean\_\theta\_Matrix$ = $sum\_\theta\_Matrix[\theta]$ / securityEventCounter;
      if $mean\_\theta\_Matrix[\theta] > max\_mean$ **then**
        max_mean = $mean\_\theta\_Matrix[\theta]$;
        $\theta_{selected} = \theta$;
      end
    end
  end
end

Algorithm 2: Q-Learning Honeypot Deployment

Data: Q(S,A), γ, $a_{learningRate}$, SE, securityEventCounter
Result: a_{action}
$\gamma = 0.9$;
$a_{learningrate}$ = 0.1;
SE = init();
$securityEventCounter = 0$;
for s **by** S **do**
  for a **by** A **do**
    $Q(s, a) = 0$;
  end
end
for $securityEventCounter \leftarrow 1$ **to** SE **by** 1 **do**
  s_t = random();
  a_t = e-Greedy();
  s_{t+1}, $R(s_t, a_t)$ = deploy(a_t);
  $TD_t(s_t, a_t) =$
  $R(s_t, a_t) + \gamma max_a(Q(s_{t+1}, a)) - Q(s_t, a_t)$
  $Q_t(s_t, a_t) =$
  $Q_{t-1}(s_t, a_t) + a_{learningRate} TD_t(s_t, a_t)$
end

VI. Deployment Wireless Honeypots in Ultra-Dense Networks

A. Communication Network Model

Let us assume a wireless network that consists of N Remote Radio Heads (RRHs), which are deployed with a Poisson point process with density λ [42], [43]. Each of the RRHs can operate either as an Access Point (AP) or a WH. Thus, at a specific time instance, L APs and M WHs are deployed with $L + M = N$, as depicted in Fig. 2. The role of the WHs is to imitate the behaviour of APs in order to attract and directly detect potential attacks. Also, we assume that the network serves K legitimate users, while the potential existence of malicious users who aim to access the real network is also considered. To mitigate their impact, the allocation of APs and WHs is dynamically adjusted and fully controlled by the network coordinator, which communicates with both the APs and WHs. However, although the WHs imitate the behaviour of APs in order to attract potential attacks, they do have access to the real network. Also, it is assumed that a normal user will never attempt to access a WH, while a malicious user may try to access either an AP or a WH. Hereinafter, let $L = (1-\theta)N$ and $M = \theta N$. Moreover, we assume that the APs follow a Poisson point process (PPP) with density $(1-\theta)\lambda$, while the WHs also follow a PPP with density $\theta\lambda$. Moreover, similarly to the scenario that has been considered in the former section, the *Attacker* attacks a specific RRH with probability ϕ. In the considered setup, the density of WHs deployment needs to be specified in order to provide the required level of security, without degrading the quality of service that is offered by the wireless communication network.

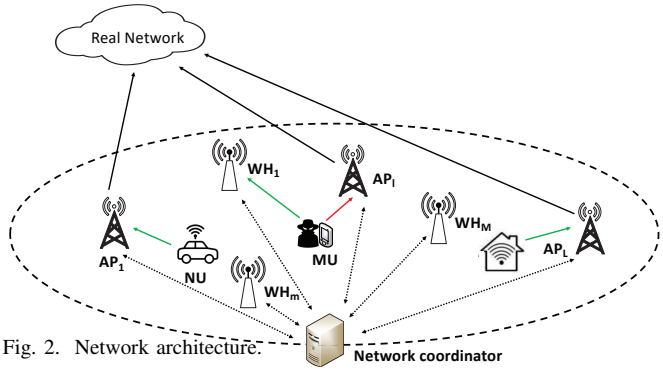

Fig. 2. Network architecture.

B. Defender Utility Function

It is assumed that each user is served by the AP that is closest to the user. The Probability Density Function (PDF) of the distance d_i between user i and its closest AP is given by:

$$f_{D,i}(d) = 2\pi d\lambda(1-\theta)\exp\left(-\lambda(1-\theta)\pi d^2\right), \quad (6)$$

while the cumulative density function (CDF) is given by

$$F_{D,i}(d) = 1 - \exp\left(-\lambda(1-\theta)\pi d^2\right). \quad (7)$$

The achievable rate is given by

$$R_i = \log_2\left(1 + \frac{h_i\gamma_i}{d_i^{\beta}}\right), \quad (8)$$

where h_i denotes the small scale fading power gain and γ_i denotes the Signal-to-Noise Ratio (SNR) at the reference distance of 1 m and is given by

$$\gamma_i = \frac{P_i}{\sigma^2 L_{ref}}, \quad (9)$$

with L_{ref} being the equivalent path-loss. In addition, P_i, σ^2, and β denote the transmit power, the noise power and the path-loss exponent, respectively.

It is assumed that each smart device has N transmission opportunities within an hour to report its demand for the next hour. Assuming that the transmission rate is equal to $R_{t,i}$, the outage probability after Z re-transmission can be defined as

$$P_{i,\text{out}} = \mathcal{P}[R_i < R_{t,i}]^Z, \quad (10)$$

where, by following similar steps as in [42],

$$\mathcal{P}[R_i < R_{t,i}] = \mathcal{P}\left[d_i \geq (\frac{h_i\gamma_i}{2^{R_{t,i}}-1})^{\frac{1}{\beta}}\right] = \int_0^{\infty}\left(1 - F_{D,i}\left(\left(\frac{h\gamma_i}{2^{R_{t,i}}-1}\right)^{\frac{1}{\beta}}\right)\right) f_{H,i}(h)dh, \quad (11)$$

with $f_{H,i}(h)$ being the PDF of the small scale fading power gain. By assuming Rayleigh fading, h_i follows the exponential distribution with parameter κ. Thus, (11) can be written as

$$P[R_i < R_{t,i}] = \int_0^{\infty}\exp\left(-\lambda(1-\theta)\pi\left(\frac{h\gamma_i}{2^{R_{t,i}}-1}\right)^{\frac{2}{\beta}}\right)\kappa\exp(-\kappa h)dh. \quad (12)$$

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2022.3184112

Hereinafter, it is assumed that $\beta = 4$, for which (11) can be written as

$$P[R_i < R_t] = 1 - \frac{\pi^{3/2}(1-\theta)\lambda\sqrt{\frac{\gamma_i}{2^{R_{t,i}}-1}}\, e^{\frac{\pi^2\gamma_i(1-\theta)^2\lambda^2}{4\kappa\left(2^{R_{t,i}}-1\right)}}\, \text{erfc}\left(\frac{\pi(1-\theta)\lambda\sqrt{\frac{\gamma_i}{2^{R_{t,i}}-1}}}{2\sqrt{\kappa}}\right)}{2\sqrt{\kappa}}. \tag{13}$$

When some nodes operate as WHs, more potential attacks are captured, however the density of nodes that operate as APs reduces, which in turn leads to an increase of the outage probability, and thus, the estimation error-cost.

Taking into account this trade-off, the aim of the *Defender* is to maximize its utility, given by (3), in which $\tilde{g}(\theta)$ is the expected overall cost due to the outage events. In more detail, $\tilde{g}$ is related to the induced cost to each user, denoted by C_i, and can be expressed as

$$\tilde{C}(\theta) = \sum_{i\in\mathcal{K}} \mathbb{E}\left[C_i\right] P_{i,\text{out}}, \tag{14}$$

with $\mathcal{K}$ being the set of users.

C. Cost of Outage Events

Indicatively, to give further insight into the definition of $\tilde{g}$ in real-world applications, the case of the smart grid can be considered, in which the cost might be related to the impact of outage events on dynamic energy management or to the case of equipment failure. To this end, next, an example that is well-known from the existing literature will be considered, which is related to Dynamic Energy Management (DEM). Assuming that the DEM operation is implemented over two consecutive stages, the unit-commitment and economic-dispatch stages, the utility generates and reserves the energy supply based on the estimated energy demand of the consumers. Thus, if the energy supply is over-estimated, the utility needs to pay for the surplus of energy that has been unnecessarily reserved with price p_{uc}. On the other hand, if the energy supply is under-estimated, the utility needs to buy the energy difference between the actual and the generated energies in the economic-dispatch stage to prevent the under-supply situation [44]. In this case, the expected cost of under or overestimating the energy demand of the devices that did not successfully report their demand is given by [44]–[46]

$$\mathbb{E}\left[C_i\right] = p_{\text{uc}} \int_0^{\mu_i} (\delta_i - r) f_{R,i} dr + p_{\text{ed}} \int_{\delta_i}^{E_{\max}} (r - \mu_i) f_{R,i} dr, \tag{15}$$

where $f_{\text{R},i}$ is the probability density function of the actual energy consumption, μ_i is the mean energy demand of the i-th device, $E_{\max}$ is the maximum energy consumption, and p_{uc} and p_{ed} are the energy prices in the unit commitment and economic-dispatch stages, respectively.

$$\mathbb{E}\left[C_i\right] = p_{\text{uc}} \int_0^{\mu_i} (\delta_i - r) f_{R,i} dr + p_{\text{ed}} \int_{\delta_i}^{E_{\max}} (r - \mu_i) f_{R,i} dr, \tag{16}$$

where $f_{\text{R},i}$ is the PDF of the actual energy consumption, μ_i is the mean energy demand of the i-th device, $E_{\max}$ is the maximum energy consumption, and p_{uc} and p_{ed} are the energy prices in the unit commitment and economic-dispatch stages, respectively.

VII. Evaluation Analysis

This section focuses on evaluating the proposed RL honeypot deployment methods: (a) $e-Greedy$ and (b) $Q-learning$ with respect to the number of WHs in ultra-dense networks. To the best of our knowledge, this is the first work related to honeypots in ultra-dense networks. Therefore, there are not publicly available datasets that can be used in the context of the evaluation analysis. To this end, we are going to use the Honeypot Intrusion Detection Dataset of our previous work in [35]. Furthermore, it is noteworthy that the e-Greedy method of [35] was appropriately adjusted in the context of this work based on the parameters of the ultra-dense networks. The aforementioned dataset includes network traffic data and relevant network flow statistics over one year from various research honeypots. This kind of data was used to create a simulation environment, identifying the values of δ_1, δ_2, δ_3, P_r, N, e and $\tilde{C}(\theta)$ given the communication network model of subsection VI-A. Since the dataset of our work in [35] is related to smart electrical systems, it can be utilised in the context of this work, taken into consideration the modelling and assumptions of section VI. Moreover, since this dataset is ready, the various security events occur by one second. Each network flow of the dataset corresponds to a security event. Thus, for each security event, we consider how many WHs will be deployed.

We consider a simulation environment where $N = 6$. Regarding the other parameters, various values of them were checked during our experiments. Therefore, we can deploy up to six WHs based on the available APs. First, with respect to the $e-Greedy$ method, we investigate how the PDF of $U_D[t]$ ranges based on Equation 2. Fig. 5-Fig. 14 show how the PDF of $U_D[t]$ ranges based on 5, 10, 20, 50, 100, 200, 500, 1000, 1500 and 2000 security events. After 2000 security events, we see that the best option is to deploy 2 WHs. Moreover Fig. 3 shows the accuracy of the e-greedy model with respect to the number of the various security events and random choice. Although due to randomness, it seems that the accuracy of the random model increases, $e-Greedy$ achieves a better accuracy. Finally, Fig. 4 shows the cumulative reward based on the iterations of $Q-Learning$ for 2000 security events.

VIII. Conclusions

The evolution of the 5G technology has led IoT and IIoT applications to the 5G era. However still security issues remain. In this paper, we investigate the use of WHs in ultra-dense networks. In particular, first, we introduce a strategic honeypot deployment method, taking full advantage of two RL methods, namely (a) $e-Greedy$ and (b) $Q-Learning$. The deployment process is converted into a MAB problem with the goal to deploy the optimal number of WHs in an ultra-dense environment, taking into account the costs and benefits of the $Defender$. The evaluation results demonstrate the efficiency of the proposed methods. Our future work will

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2022.3184112

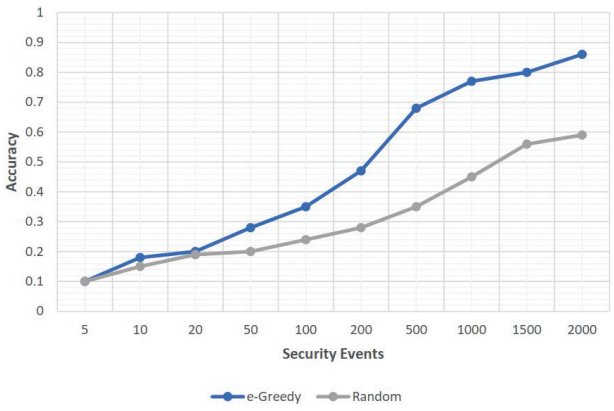

Fig. 3. e-Greedy Accuracy vs Random Choice

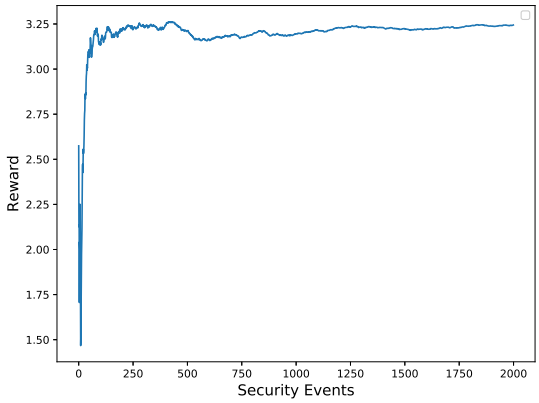

Fig. 4. Q-Learning Reward per Security Events

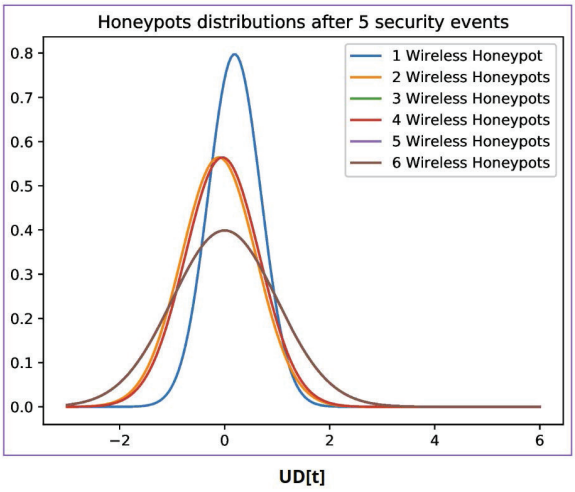

Fig. 5. Honeypots Distribution after 5 sec. events

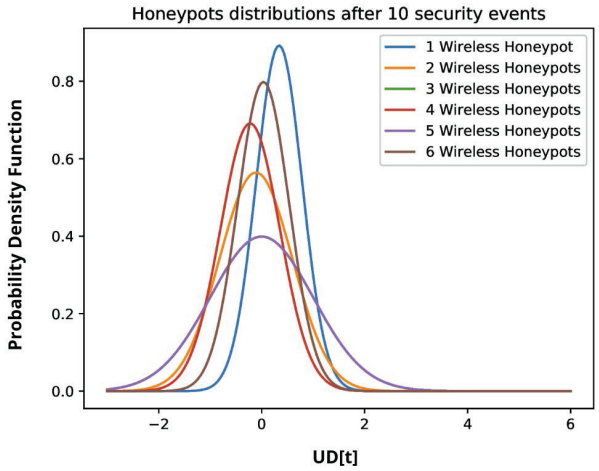

Fig. 6. Honeypots Distribution after 10 sec. events

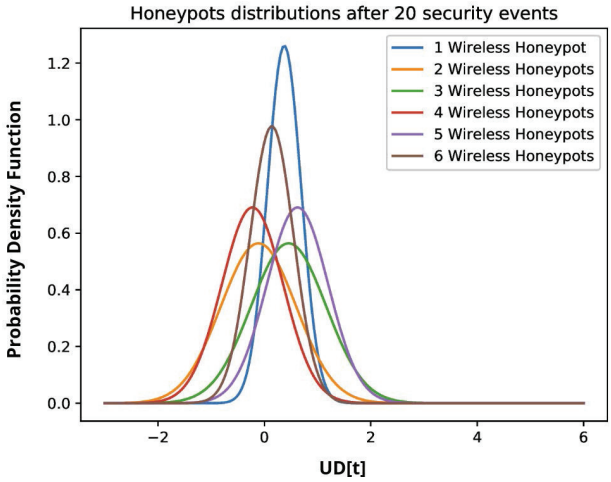

Fig. 7. Honeypots Distribution after 20 sec. events

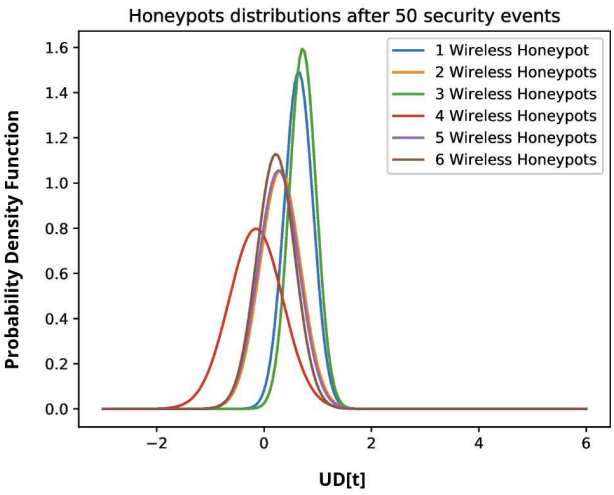

Fig. 8. Honeypots Distribution after 50 sec. events

This article has been accepted for publication in IEEE Transactions on Emerging Topics in Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TETC.2022.3184112

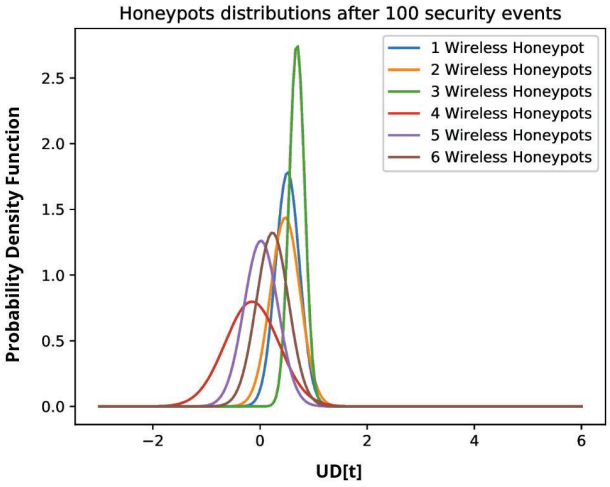

Fig. 9. Honeypots Distribution after 100 sec. events

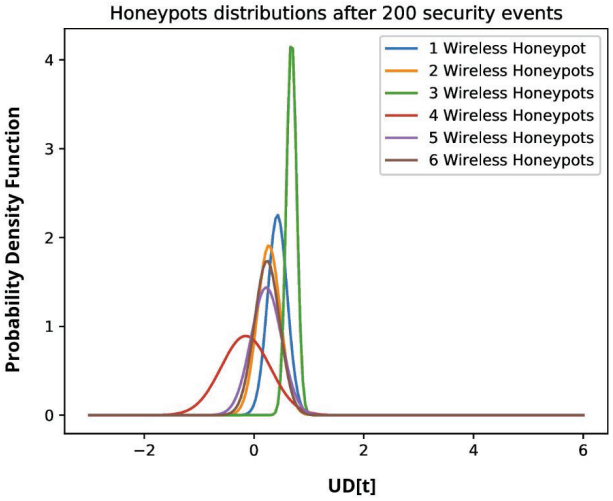

Fig. 10. Honeypots Distribution after 200 sec. events

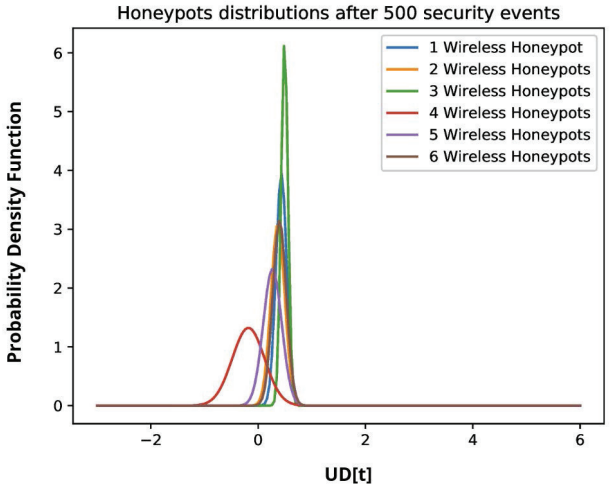

Fig. 11. Honeypots Distribution after 500 sec. events

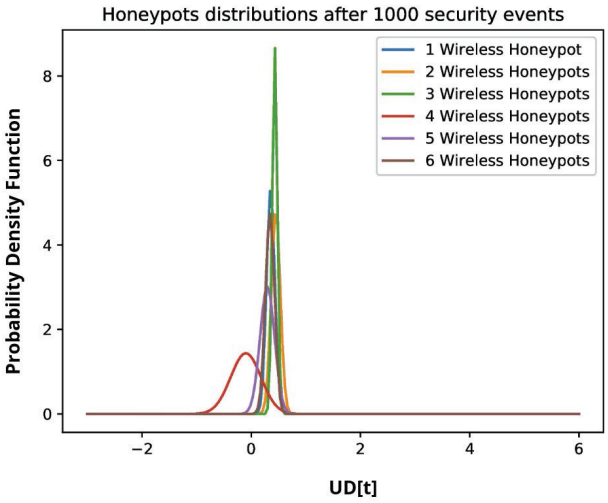

Fig. 12. Honeypots Distribution after 1000 sec. events

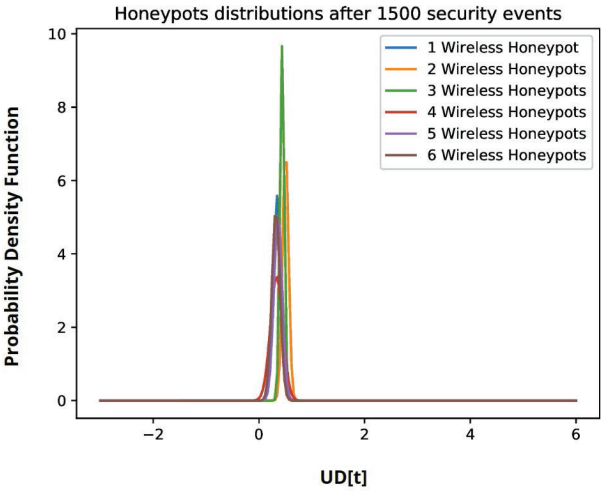

Fig. 13. Honeypots Distribution after 1500 sec. events

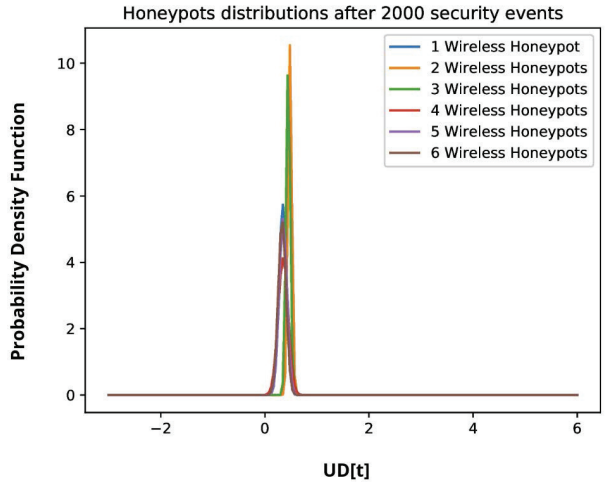

Fig. 14. Honeypots Distribution after 2000 sec. events

focus on investigating more complex RL techniques for using WHs in the 5G-RAN, 5G Core and B5G networks.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 833955 (SDN-microSENSE).

REFERENCES

[1] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, "Overview of 5g security challenges and solutions," *IEEE Communications Standards Magazine*, vol. 2, no. 1, pp. 36–43, 2018.

[2] D. B. Rawat and S. R. Reddy, "Software defined networking architecture, security and energy efficiency: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 325–346, 2016.

[3] M. Pattaranantakul, R. He, Q. Song, Z. Zhang, and A. Meddahi, "Nfv security survey: From use case driven threat analysis to state-of-the-art countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3330–3368, 2018.

[4] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5g technologies: Potential solutions, recent advancements, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 196–248, 2019.

[5] J. Cao, M. Ma, H. Li, R. Ma, Y. Sun, P. Yu, and L. Xiong, "A survey on security aspects for 3gpp 5g networks," *IEEE communications surveys & tutorials*, vol. 22, no. 1, pp. 170–195, 2019.

[6] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *Journal of Network and Computer Applications*, vol. 169, p. 102767, 2020.

[7] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, "Securing the smart grid: A comprehensive compilation of intrusion detection and prevention systems," *IEEE Access*, vol. 7, pp. 46 595–46 620, 2019.

[8] M. H. Almeshekah and E. H. Spafford, "Cyber security deception," in *Cyber deception*. Springer, 2016, pp. 23–50.

[9] C. Irvene, D. Formby, S. Litchfield, and R. Beyah, "Honeybot: A honeypot for robotic systems," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 61–70, 2017.

[10] D. Kumar and V. Vashishtha, "Snort based h-ids with kf sensor and weka," *International Journal*, vol. 2, no. 5, 2012.

[11] M. Tsikerdekis, S. Zeadally, A. Schlesener, and N. Sklavos, "Approaches for preventing honeypot detection and compromise," in *2018 Global Information Infrastructure and Networking Symposium (GIIS)*. IEEE, 2018, pp. 1–6.

[12] M. Nawrocki, M. Wählisch, T. C. Schmidt, C. Keil, and J. Schönfelder, "A survey on honeypot software and data analysis," *CoRR*, vol. abs/1608.06249, 2016. [Online]. Available: http://arxiv.org/abs/1608.06249

[13] V. Sethia and A. Jeyasekar, "Malware capturing and analysis using dionaea honeypot," in *2019 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2019, pp. 1–4.

[14] R. K. Shrivastava, B. Bashir, and C. Hota, "Attack detection and forensics using honeypot in iot environment," in *International Conference on Distributed Computing and Internet Technology*. Springer, 2019, pp. 402–409.

[15] S. Gokhale, A. Dalvi, and I. Siddavatam, "Industrial control systems honeypot: A formal analysis of conpot." *International Journal of Computer Network & Information Security*, vol. 12, no. 6, 2020.

[16] R. Siles, "Honeyspot: The wireless honeypot," *Spanish Honeynet Project*, pp. 1–28, 2007.

[17] N. Al-Gharabally, N. El-Sayed, S. Al-Mulla, and I. Ahmad, "Wireless honeypots: survey and assessment," in *in Proc. of the 2009 conference on Information Science, Technology and Applications*, 2009, pp. 45–52.

[18] M. Gowri and B. Paramasivan, "Rule-based anomaly detection technique using roaming honeypots for wireless sensor networks," *ETRI Journal*, vol. 38, no. 6, pp. 1145–1152, 2016.

[19] I. Koniaris, G. Papadimitriou, and P. Nicopolitidis, "Analysis and visualization of ssh attacks using honeypots," in *Eurocon 2013*. IEEE, 2013, pp. 65–72.

[20] A. Tiwari and D. Kumar, "Comparitive study of various honeypot tools on the basis of their classification & features," *Available at SSRN 3565078*, 2020.

[21] S. Manchekar, M. Kadam, and K. Jamdaade, "Application of honeypot in cloud security: A review," *International Journal on Future Revolution in Computer Science & Communication Engineering*, vol. 4, no. 6, pp. 63–65, 2018.

[22] A. K. Seewald and W. N. Gansterer, "On the detection and identification of botnets," *Computers & Security*, vol. 29, no. 1, pp. 45–58, 2010.

[23] C. Leita, V. Pham, O. Thonnard, E. Ramirez-Silva, F. Pouget, E. Kirda, and M. Dacier, "The leurre.com project: Collecting internet threats information using a worldwide distributed honeynet," *2008 WOMBAT Workshop on Information Security Threats Data Collection and Sharing*, 2008.

[24] G. Portokalidis, A. Slowinska, and H. Bos, "Argos," *Proceedings of the 2006 EuroSys conference on - EuroSys 06*, 2006.

[25] P. Radoglou-Grammatikis, P. Sarigiannidis, E. Iturbe, E. Rios, A. Sarigiannidis, O. Nikolis, D. Ioannidis, V. Machamint, M. Tzifas, A. Giannakoulias *et al.*, "Secure and private smart grid: The spear architecture," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2020, pp. 450–456.

[26] P. Radoglou Grammatikis, P. Sarigiannidis, C. Dalamagkas, Y. Spyridis, T. Lagkas, G. Efstathopoulos, A. Sesis, I. L. Pavon, R. T. Burgos, R. Diaz *et al.*, "Sdn-based resilient smart grid: The sdn-microsense architecture," *Digital*, vol. 1, no. 4, pp. 173–187, 2021.

[27] A. Uprety and D. B. Rawat, "Reinforcement learning for iot security: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8693–8706, 2020.

[28] J. Franco, A. Aris, B. Canberk, and A. S. Uluagac, "A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2351–2383, 2021.

[29] W. Fan, Z. Du, D. Fernández, and V. A. Villagrá, "Enabling an anatomic view to investigate honeypot systems: A survey," *IEEE Systems Journal*, vol. 12, no. 4, pp. 3906–3919, 2017.

[30] C. Dalamagkas, P. Sarigiannidis, D. Ioannidis, E. Iturbe, O. Nikolis, F. Ramos, E. Rios, A. Sarigiannidis, and D. Tzovaras, "A survey on honeypots, honeynets and their applications on smart grid," in *2019 IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2019, pp. 93–100.

[31] C. Kiekintveld, V. Lisỳ, and R. Píbil, "Game-theoretic foundations for the strategic use of honeypots in network security," in *Cyber warfare*. Springer, 2015, pp. 81–101.

[32] L. Shi, Y. Li, and H. Feng, "Performance analysis of honeypot with petri nets," *Information*, vol. 9, no. 10, p. 245, 2018.

[33] W. Zhang, B. Zhang, Y. Zhou, H. He, and Z. Ding, "An iot honeynet based on multiport honeypots for capturing iot attacks," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3991–3999, 2019.

[34] L. Shi, Y. Li, T. Liu, J. Liu, B. Shan, and H. Chen, "Dynamic distributed honeypot based on blockchain," *IEEE Access*, vol. 7, pp. 72 234–72 246, 2019.

[35] P. Radoglou-Grammatikis, A. Liatifis, E. Grigoriou, T. Saoulidis, A. Sarigiannidis, T. Lagkas, and P. Sarigiannidis, "Trusty: A solution for threat hunting using data analysis in critical infrastructures," in *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2021, pp. 485–490.

[36] P. Diamantoulakis, C. Dalamagkas, P. Radoglou-Grammatikis, P. Sarigiannidis, and G. Karagiannidis, "Game theoretic honeypot deployment in smart grid," *Sensors*, vol. 20, no. 15, p. 4199, 2020.

[37] K. Wang, M. Du, S. Maharjan, and Y. Sun, "Strategic honeypot game model for distributed denial of service attacks in the smart grid," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2474–2482, 2017.

[38] Y. Zhang, C. Di, Z. Han, Y. Li, and S. Li, "An adaptive honeypot deployment algorithm based on learning automata," in *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2017, pp. 521–527.

[39] M. Du and K. Wang, "An sdn-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 648–657, 2019.

[40] U. Bartwal, S. Mukhopadhyay, R. Negi, and S. Shukla, "Security orchestration, automation, and response engine for deployment of behavioural honeypots," *arXiv preprint arXiv:2201.05326*, 2022.

[41] W. Fan, Z. Du, M. Smith-Creasey, and D. Fernandez, "Honeydoc: an efficient honeypot architecture enabling all-round design," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 683–697, 2019.

[42] T. D. Novlan, H. S. Dhillon, and J. G. Andrews, "Analytical modeling of uplink cellular networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 6, pp. 2669–2679, 2013.

[43] A. I. Aravanis, O. Muñoz, A. Pascual-Iserte, and J. Vidal, "Analysis of downlink and uplink decoupling in dense cellular networks," in *2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, 2016, pp. 219–224.

[44] A. E. Shafie, H. Chihaoui, R. Hamila, N. Al-Dhahir, A. Gastli, and L. Ben-Brahim, "Impact of passive and active security attacks on MIMO smart grid communications," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2873–2876, 2019.

[45] A. El Shafie, D. Niyato, R. Hamila, and N. Al-Dhahir, "Impact of the wireless network's phy security and reliability on demand-side management cost in the smart grid," *IEEE Access*, vol. 5, pp. 5678–5689, 2017.

[46] D. Niyato, P. Wang, and E. Hossain, "Reliability analysis and redundancy design of smart grid wireless communications system for demand side management," *IEEE Wireless Communications*, vol. 19, no. 3, pp. 38–46, 2012.

Panagiotis Radoglou-Grammatikis received the Diploma degree from the Dept. of Informatics and Telecommunications Eng. (now Dept. of Electrical and Computer Eng.), Faculty of Eng., University of Western Macedonia, Greece, in 2016. He is now a PhD candidate in the same department. His main research interests are in the area of cybersecurity and mainly focus on intrusion detection, vulnerability research and applied cryptography. He has published more than 20 research papers in international scientific journals, conferences and book chapters. He has served as a reviewer for several scientific journals and possesses working experience as a security engineer and software developer. Currently, he is working as a research associate at the University of Western Macedonia in national and European funded research projects. Finally, he participates in the topic board of Electronics (MDPI publishing) and he is a member of IEEE and the Technical Chamber of Greece.

Panagiotis Sarigiannidis is the Director of the ITHACA Lab and Associate Professor at the Department of Electrical and Computer Engineering at the University of Western Macedonia, Kozani, Greece. He received the B.Sc. and Ph.D. degrees in computer science from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2001 and 2007, respectively. He has published over 200 papers in international journals, conferences and book chapters. He has been involved in several national, European and international projects. He is currently the project coordinator of three H2020 projects, namely SPEAR, EVIDENT and TERMINET. He also coordinates MARS and the Erasmus+ KA2 ARRANGE-ICT: SmartROOT. He serves as a principal investigator in H2020 SDN-microSENSE, and in three Erasmus+ KA2: ARRANGE-ICT, JAUNTY and STRONG. His research interests include telecommunication networks, IoT and network security. He is an IEEE member and participates in the Editorial Boards of various journals.

Panagiotis D. Diamantoulakis received the Diploma (five years) and PhD from the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki (AUTH), Greece, in 2012 and 2017, respectively. Since 2017, he works as a Post-doctoral Fellow in Wireless Communications & Information Processing (WCIP) Group at AUTH and, since 2021, he is also a visiting Assistant Professor in the Key Lab of Information Coding and Transmission at Southwest Jiaotong University (SWJTU), China. From 2018 to 2020, he also worked as visiting Postdoctoral Researcher in the Key Lab of Information Coding and Transmission at SWJTU and in the Institute for Digital Communications (IDC) of the Telecommunications Laboratory (LNT) at Friedrich-AlexanderUniversitat Erlangen-Nurnberg (FAU), Germany. His current research ¨ interests include resource allocation in wireless communications, optimization theory and applications in wireless networks and smart grids, game theory, wireless power transfer and optical wireless communications.

Thomas Lagkas is Assistant Professor of the Department of Computer Science of the International Hellenic University. He received his PhD in computer science from the Aristotle University of Thessaloniki, Greece, in 2006. He has been Lecturer and then Senior Lecturer of The University of Sheffield International Faculty - CITY College, from 2012 to 2019. He also served as Research Director of the Computer Science Department of CITY College and Leader of the ICT Track of the South-East European Research Centre. His research interests are in the broad area of IoT communications with more than 90 publications at widely recognized international scientific journals and conferences. Dr. Lagkas is Fellow of the Higher Education Academy in UK. He also participates in the Editorial Boards of respectful scientific journals and is actively involved in drafting research funding proposals, as well as in the implementation of EU projects.

Theocharis Saoulidis received his B.Sc. in "Informatics" (2019) and his M.Sc in "Web Intelligence"(2021) from Alexander TEI of Thessaloniki (Thessaloniki, Greece). He participated in Erasmus+ programme and he visited the University of Cagliari and during this period he participated in many research projects related to 4G/5G, Quality of Experience (QoE) and Software Defined Networks (SDN). He has published research papers in important conferences such as IEEE ICC. His major research interests include Machine Learning, Software Defined Networks, Network function virtualization, Internet-of-Things and Cybersecurity. He participates as a software engineer for national and EU-funded projects.

Eleftherios Fountoukidis is a scientific researcher in the field of information security. He received the B.Sc. degree from the Dept. of Applied Informatics and Multimedia of the Technological Educational Institute of Crete. His research interests include information security, web applications and Internet-of-Things. He is with SID since 2021.

Prof. George K. Karagiannidis (M'96, SM'03, F'14) received the University Diploma (5 years) and PhD degree, both in electrical and computer engineering from the University of Patras, in 1987 and 1999, respectively. He is currently Professor in the Electrical & Computer Engineering Dept. and Head of Wireless Communications Systems Group (WCSG). He is also Honorary Professor at South West Jiaotong University, Chengdu, China. His research interests are in the broad area of Digital Communications Systems and Signal processing. Currently, he serves as Associate Editor-in Chief of IEEE Open Journal of Communications Society. He is one of the highly-cited authors across all areas of Electrical Engineering, recognized from Clarivate Analytics as Web-of-Science Highly-Cited Researcher during 2015-2021. Prof. Karagiannidis received the 2021 IEEE Communications Society Radio Communications Committee Technical Recognition Award and the 2018 Signal Processing and Communications Electronics Technical Recognition Award of the IEEE Communications Society.