Intelligence at the Edge of Complex Networks: the case of cognitive transmission power control

Pasquale Pace[1]*, Giancarlo Fortino*, Yin Zhang † Antonio Liotta ‡,
* Department of Informatics, Modeling, Electronics and Systems
University of Calabria - Rende, Italy
Email: p.pace@dimes.unical.it - g.fortino@unical.it
† Zhongnan University of Economics and Law, China
E-mail: yinzhang@zuel.edu.cn
‡ Data Science Research Centre
University of Derby, UK
E-mail: a.liotta@derby.ac.uk

***Abstract*—The rapid proliferation of new devices has led to the Internet of Things (IoT), a network where virtually any object equipped with a radio interface could get connected. Accordingly, networks are exploding in terms of the number of devices but also in complexity. The key issue arises from the increasing density in wireless communications, which the "*deterministic*" nature of current protocols can no longer handle. Herein, we explore ways in which the latest development in artificial intelligence (AI) and, particularly, machine learning (ML) may help addressing the complex requirements of IoT communications, highlighting the crucial role of "*predictive*" communications. We illustrate the software architectures and the fundamental mechanisms that can enable AI processes in communications. Finally, we introduce an exemplary case study where machine learning is successfully used to find the delicate balance between spectrum and energy efficiency in wireless sensor networks. The emerging panorama for cognitive communications is one in which intelligent processes must start at the very edge and need to transfer meta-learned information in a peer-to-peer fashion.**

***Index Terms*—Cognitive communications, Intelligent Networks, Machine Learning, Transfer Learning, Internet of Things, Wireless Sensor Networks.**

I. Introduction

The widespread digitization of the physical world and the Internet of Things (IoT) trend to connect virtually any object equipped with a radio interface, are creating ever more complex systems. Across all sectors of industry and society, there is a quest to adopt Artificial Intelligence (AI) to master the complexity of IoT systems, which requires pervasive, wireless connectivity beyond what is currently possible. While AI is being used broadly in the most innovative applications and services, there is an opportunity to use those very same intelligent mechanisms to construct more intelligent and cognitive communications[1]. The key issue with pervasive connectivity is not only the shear number of devices (projected to be at the tunes of trillions) but the increasing density in wireless communications, which the "deterministic" nature of current protocols can no longer handle.

As the number of commercial and industrial IoT devices proliferates, getting such a diverse range of "things" to behave and connect intelligently is crucial to realizing the full potential of our digital ecosystems. An industrial factory typically uses thousands of sensors to monitor the status of thousands of machines and processes. Yet conventional communication protocols are not geared for such industrial IoT requirements, where reliable wireless connectivity and ultra-low latency are a must. In most challenging IoT applications, sensor data needs to be collected, analyzed and correlated with historical performance data to make decisions in real-time [2]; as an example you can consider swarms of industrial drones or remote facilities filled with smart sensors and actuators that need to communicate and coordinate with each other to accomplish tasks without being connected to a remote AI cloud service [3],[4].

In such cases, embedding *intelligence* at the source of the sensing is the most sensible way to obtain timely actionable reactions. What is more, the very same intelligent mechanisms may be used both at the application level (*e.g.*, to detect anomalies or predict trajectories) and to meet the demands of real-time communications (*e.g.*, enabling prediction-based protocols).

The opportunities and challenges to exploit AI to achieve intelligent 5G networks by effectively orchestrating cellular network resources, have been widely investigated in the last few years [5]. However, the time has come to focus on future designs and working paradigms consisting in moving the intelligence from the *core* to the *edge* of the network[6], to enable decisions to be made directly on the IoT nodes (*i.e.*, on resource constrained sensors), rather than "*phoning home*" to headquarters or a cloud service to find out "*what to do next*".

Thanks to recent breakthroughs in lightweight AI methods, it is now possible to bring IoT systems to the next level, allowing localized (rather than cloud-centric) decisions and minimizing the communication footprint. That will, in turn, allow enterprises to better capitalize on their IoT investments.

[1] corresponding author

It is crucial for IoT devices to be able to work together, for instance to diagnose and solve problems in real-time, even in disconnection from central big-data capability.

Starting from this challenging vision of decentralized intelligence, here we explore the various ways in which the latest AI developments, particularly machine learning, may help addressing the complex requirements of IoT communications, highlighting the crucial role of "predictive" rather than "deterministic" communications. First, we illustrate the software architectures that may be used for moving intelligence towards the edge and transferring meta-learned models (Sect. II). Next, we look at learning mechanisms that can naturally be deployed in the IoT node (Sect. III). Finally, we introduce an exemplary case study where machine learning is successfully used to find the delicate balance between spectrum and energy efficiency in wireless sensor network.

This level of autonomy and self-adaption (through machine learning) within individual nodes is crucial in a context that is continuously changing in ways that conventional "deterministic" models fail to capture. In our prototyping study we wanted to ascertain the extent by which lightweight machine learning could beneficially be used in cognitive communications. Our findings, based on a range of diverse sensors (some having as little as 20kbytes of memory), show great potential and encourage further studies in this direction. However, in this work we only present the results related to a specific sensor platform due to the space limits.

II. General Software Architectures for Learning

Since new attractive applications have caused the data volume in both fixed and mobile networks to skyrocket in recent years, it is foreseeable that the technologies available today will be unable to satisfy users' demands for service quality in the future. In this context, *cognitive wireless technologies* play an important role when it comes to reliability of communication often setting stringent requirements on the energy efficiency and robustness of the wireless systems, which makes a type of "on-the-fly" adjustment to the residual energy level necessary in practical implementation.

Artificial intelligence and, particularly, machine learning have considerably evolved in the past decade, with a broad set of methods and applications. Our intention in this section is to explore which general software architectures for learning (*i.e.,* Traditional Vs. Distributed shown in Fig. 1) may be employed in cognitive wireless communications to support the well know cognitive cycle in which the radio should be able to learn from its past actions.

A. Centralized Learning

Up until recently, the predominant method for employing machine learning has been the centralized one. In the Internet of Things context, this has been referred to as cloud-assisted sensing, whereby all contextual data needs to be transported onto a central point before it can be used to make prediction models and inference. The sensor nodes are "*dumb*" in the sense that their functional requirements are limited to data collection and communication. All intelligence resides in servers, which is where sufficient computing resources are concentrated. Centralized learning shares the pros and cons of centralization; on the one hand, it can rely on rich datasets and powerful computations. Yet, communication becomes the limiting factor, particularly considering the projected scale of IoT systems and their predominant use of wireless spectrum.

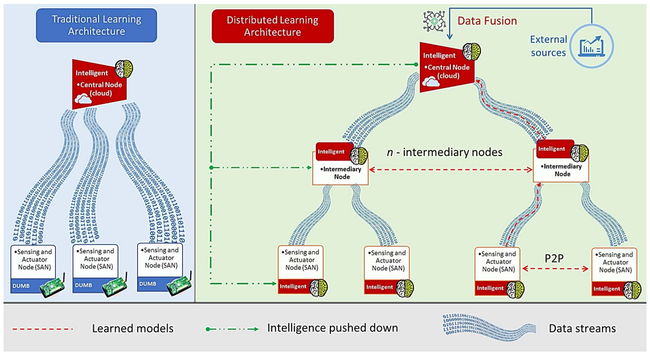


Fig. 1. Traditional learning architecture Vs Distributed learning architecture.

B. In-node Learning

Recent development in AI has made it possible to miniaturize machine learning processes to the extent that online learning can be run onto limited devices, with as little memory as a few dozens kilobytes [7]. Thanks to new breakthroughs at the intersection between network science and machine learning, it is now possible to train artificial neural networks (ANN) directly onto constrained devices, rather than in powerful servers [8]. It is therefore realistic to conceive that intelligent processes should start at the network periphery, leading to new forms of decentralized machine learning. Decentralized learning opens several new avenues when it comes to cognitive wireless communications, as it allows wireless devices to directly spot communication patterns without incurring any communication overheads with base stations or other centralized points [9]. In-node learning has significant potential to evolve current communication standards towards prediction-based communications at the physical, MAC and network layers, and enabling cross-layer communication protocols for high-density, low-power, massive-scale systems.

C. Transfer Learning

In-node learning represents a major step forward compared to a centralized architecture, and is also the starting point for even more elaborate forms of decentralized intelligence. With transfer learning it is possible to transfer information learned across nodes in a peer-to-peer learning fashion [10]. The value to intelligent communications is unprecedented, as it allows for a learned node to bootstrap or accelerate the learning transient of another one. High-level, meta information can travel not only from top to bottom (*e.g.*, from the cloud down to individual nodes) but also between similar nodes (*e.g.*, among neighboring sensors). It is also conceivable to use transfer learning to generate meta-learning models, starting from sub-domains. All these possibilities are very important in cognitive communications, where it is not always possible for a device to learn communication patterns (*e.g.*, due to lack of access to data or limited computing capability). An exemplary work on the use of transfer learning as a mechanism for cooperative management in cognitive radio has been carried out by Zhao *et al.* [11].

III. Learning Mechanisms for Constrained Devices

Thanks to recent breakthroughs in machine learning miniaturization, it is now possible to contemplate the realization of AI tasks directly onto constrained devices, that are digital objects such as wireless sensors having limited computing capability. The key difference with other programmatic approaches is the use of online reinforcement learning mechanisms, which allow individual nodes to self-tune, without requiring any predetermined rules or threshold setting [12]. To this end, cognitive IoT devices can use local information to find out the communication and interference patterns and make prediction models. These can be used to drive transmission power and channel selection so as to minimize packet collision and energy consumption, without having to tightly rely on cloud resources or on edge-to-cloud communication. Next, we explore three key mechanisms that can make real impact on cognitive communications.

A. Anomaly Detection

IoT systems are typically used to gather vast amounts of data, with a view to identify anomalies (such as system failures, intrusion, or unanticipated behavior of the environment) or other events of interest and take appropriate actions, accordingly. Using conventional AI to detect, prevent or adjust to events means that the whole of data gathered at the edge would need to be transferred to the cloud, before any process may be applied. This is unfeasible, considering the scale and resource constraints of IoT systems. Specific problems include: *i)* limited wireless spectrum around devices; *ii)* limited energy available on the devices that are not typically able to permanently stay in transmit mode; *iii)* limited computing available on the server side, considering that IoT devices are projected to grow by 2-3 orders of magnitude within the next few years. Lightweight online and reinforcement learning methods come to the rescue. Bosman *et al.* [7] have done extensive work to evaluate the viability of various ML methods in tiny sensors. Their work proves that non-trivial events and diverse anomalies may be picked up on-the-fly and without having to rely on any prior knowledge. That means that, despite the limitations of embedded platforms, it is possible to provide them with sufficient intelligence as to self-adjust their transmission parameters based on the observed context. That may include information from any of the network layers and including also the data patterns. What is even more potential, is the ability to fuse or transfer meta-learned information among neighboring devices to further improve the detection of anomalies (in terms of precision and recall) [13]. Anomaly detection at the edge is a fundamental building block in edge intelligence, as it allows for substantial filtering and compression of data, which is paramount to scaling up IoT communications.

B. Resource Management

Edge devices, sensors, actuators, controllers and other sort of embedded systems are typically battery powered and communicate through wireless protocols. Efficient resource management is therefore of paramount importance, particularly finding optimal transmission frequencies and power and minimizing packet collision and latency. These are conflicting requirements which demand going well beyond the simple, reactive protocols in use. AI offers the opportunity to develop predictive methods that prevent, rather than address, issues such as packet re-transmissions.

Chincoli *et al.*[9] have carried out exemplary work on learning-based resource management in wireless sensor networks, showing how reinforcement learning may be used in small sensors to tackle the combined problems of energy and spectrum efficiency. Q-learning can indeed be deployed in low-spec devices to iteratively predict the channel conditions through *trial-and-error* methods that are more viable than other optimization algorithms (Sections IV and V).

Overall, in-node reinforcement learning is a pre-requisit for typical IoT devices to continuously strive for minimum energy status.

C. Self-characterization

Online reinforcement learning has crucial applications in smart sensing and predictive communications, as illustrated in the aforementioned examples. However, the true power of AI resides in the ability to train artificial neural networks (ANN) based on sample datasets that reflect the context of the given application. Meaningful ANNs require considerable computing power, which is typically available only in the cloud. This has so far confined ANNs to centralized learning architectures (Sect. II.A), whereby the ANN is first trained in the cloud and then pushed down to the edge node.

This form of intelligence decentralization is effective in various applications, but still requires considerable batch communication and processing between the many IoT nodes and a few cloud service centers. There is still a centralized bottleneck in the process, which limits the real-time element of AI in IoT.

However, thanks to recent developments reported in [14] it is now conceivable to train ANNs directly in edge nodes. This enables new possibilities in IoT intelligence, since individual nodes will soon be able to self-characterize their contextual parameters in real-time, i.e. via limited or no communication with cloud services.

In terms of intelligent communications, in-node ANNs open the door to far more advanced prediction capability, which is necessary to capture the complex realm of wireless communications.

IV. Cognitive Transmission Power Control

In this section, we investigate one of the aforementioned ML techniques in constrained devices, to better illustrate and substantiate the "*intelligence at the edge*" vision. We describe a cognitive, cooperative Transmission Power Control (TPC) scheme based on reinforcement learning (RL) agents, whereby each wireless sensor node iteratively learns its minimum energy level. Since all sensors incorporate the same RL agent functionality, this is an example of multi-agent systems that strives for optimality through a sequence of trial-and-error actions. Specific goals are specified in terms of network quality of service (QoS) parameters. As an example, we target the combination of minimum energy and minimum packet error rate, showing that this has positive effect in terms of both energy and spectrum efficiency.

The agent is the decision maker of the system that interacts and influences the environment (*i.e.*, the wireless channel). The decisions that the agent takes are called actions, $a \in A \subseteq \mathbb{N}$, where A is a set of actions.

The status of the environment is defined as state $s \in S \subseteq \mathbb{N}$, where S is a set of states. The agent is rewarded via a discrete range of negative and positive numbers, whether the action taken provides a positive or negative effect on the environment towards the goal, respectively. The so called reward is defined as $r \in R \subseteq \mathbb{R}$, where R is a set of rewards. The TPC protocol adopts the Q-learning algorithm, a well-known lightweight RL solution [15], suitable for constrained devices by computing the long-term quality value (Q-value) of the state-action combinations (s_k, a_k), at every k-th episode. Q-value is updated by using the following formula:

$$\begin{gathered} Q_{k+1}(s_k, a_k) = (1 - l_k) * Q_k(s_k, a_k) + \\ l_k(s_k, a_k) * [r_{k+1} + \delta * max_a \; Q_k(s_{k+1}, a)] \end{gathered} \tag{1}$$

where $l_k \in U[0,1]$ is the learning factor at the k-th episode, providing the system to be between purely conservative (*i.e.*, l=0) or adaptive (*i.e.*, l=1). The parameter $\delta \in U[0,\ 1]$ is the discount factor, which weighs the contribution of the future Q-value estimations $max_a \; Q_k(s_{k+1}, a)$.

The episode k represents the transmission of N packets within a window W, using the same action. At the end of the current window, the agent is rewarded, the new state is updated and the action is changed; then, W is flushed and new packets are sent.

Q-learning is an off-policy algorithm, which means that the action selection does not follow a defined policy π but it changes over time, following a strategy. The strategy of the proposed protocol is ε-greedy, which is based on the comparison of $\varepsilon \in U[0,1]$ with a uniformly distributed random value $x \in U[0,1]$. In this way, when $x \leq \varepsilon$ the system explores; otherwise it exploits the actions that produced the maximum Q-value for the specific state s_k. The ε factor is variable in accordance with a scheduling setup related to the episode count.

In summary, the different components of the proposed Cognitive Q-Learning TPC (CQL-TPC) protocol have the following meanings: *i)* the action is the transmission power level, *ii)* the state is the interference intensity at the transmitter and receiver side, mapping both the Clear Channel Assessment (CCA) attempts and the retransmissions combination in discrete values, and *iii)* the reward takes into account the quantization of the Packet Reception Ratio (PRR) and the selected transmission power combinations (*i.e.*, the higher the PRR and the lower the transmission power, the higher the reward).

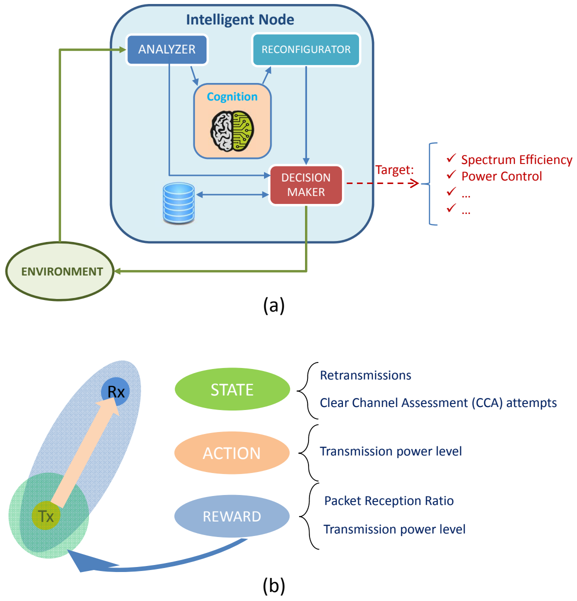


Fig. 2. a) Intelligent node, b) States, Actions and Rewards of the Intelligent node.

In multi-agent systems, each parameter is independent and specific per agent; thus the k^{th} episodes are also different and occur asynchronously. For this reason, actions may be taken at the same time by different agents, whereas other nodes are idle. Since the agents do not know the <state-action-reward> triplet of other nodes, game theory plays an important role for converging the node decisions to a near-optimal global equilibrium. In this context, the nodes are the players of a common interest theoretical game in which the payoff of each player is maximized if their decisions improve the interference mitigation for a global benefit (*i.e.*, the nodes gain higher reward by selecting lower power levels). Figure 2.a illustrates the different functional blocks composing the intelligent node and the interactions with the environment, and the figure 2.b summarizes the states, actions and rewards during the transmission process of a nodes pair.

V. Experiments and Results

This section shows the experiments conducted on a real testbed in which the multi-agent CQL-TPC strategy has been implemented in the sensor nodes. In particular, the network performance and energy consumption have been evaluated in different density conditions in terms of nodes by comparing the proposed CQL-TPC scheme with a standard homogeneous scheme, named MaxPow, in which the transmission is always performed with the maximum power level. In addition, the convergence and scalability of the proposed approach are analyzed and discussed.

A. Testbed setup and working parameters

The CQL-TPC scheme is implemented in ZigBit nodes [16] composed by *i)* the ATxmega256A3U processor with 32 MHz of computational speed, 256kB + 8kB of Flash memory and 16kB of SRAM, *ii)* the AT86RF233 low power transceiver working at 2.4 GHz which can be used to develop IEEE 802.15.4 applications and, *iii)* a USB connector.

Since many functionalities are embedded in the hardware, we expect to obtain reliable and accurate results; moreover, the CQL-TPC scheme has been programmed in the integrated development platform, *Atmel Studio*, provided by the manufacturer to support the IEEE standard 802.15.4 module. The developed software code is lightweight enough since it occupies only 20.12% of the total memory.

The sensor nodes are deployed in a specific environment to minimize the external interferences; in particular we used a Faraday cage, thus focusing only on the interference that is caused by our sensor nodes. They are placed to form a Manhattan grid symmetric topology in which we tested four transmission scenarios, depicted in figure 3 consisting of one, two, four and eight node pairs, respectively. Each transmitter sends packets to its associated receiver at d=1m distance.

As main performance indexes we measured: *i)* the Packet Reception Ratio (PRR) consisting of the ratio of the received ACKs and the generated packets, calculated over a window W of 10 packets; *ii)* the latency, defined as the difference between the reception and generation packet time, averaged among the received packets within W; *iii)* the average energy consumption of the nodes with respect to the transmission power levels chosen by the CQL-TPC algorithm.

Table I summarizes all the parameters and the related values used in the experiments.

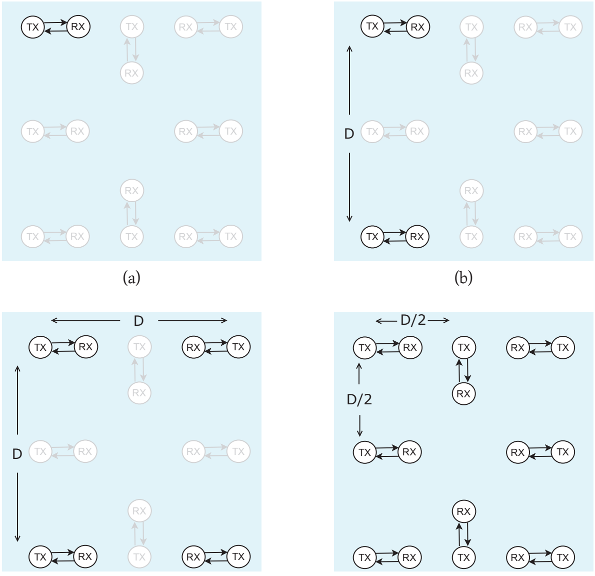
According to the implemented scenarios, the *states* s_k and *rewards* r_k are calculated as follows:

$$s_k = \|\overline{retr_k}\| + \|\overline{cca_k}\| * (n_{retr} + 1) \tag{2}$$

$$r_k = \Delta * [(prr_k^q - 1) * n_{ptx} + (n_{ptx} - p_{tx,k}) - \frac{m_r}{2}] \tag{3}$$

where $retr_k$ and cca_k are the retransmissions and the number of CCA obtained in the k-th episode, n_{retr} is the maximum number of retransmissions allowed for each packet, prr_k^q is the quantized PRR over m^{prr} levels, n_{ptx} is the number of transmission power levels, $p_{tx,k}$ is the transmission power level used during the k-th episode and m_r is the number of quantization levels for the reward, respectively.

Regarding the learning phase of the algorithm and the ε-greedy factors, we scheduled the relative values as a function of the episodes (*i.e.*, the packets transmitted in one window). The scheduling is planned according to the three phases of *learning*, *convergence* and *testing* shown in Table II. In particular, at the beginning of the learning procedure, the learning factor l is kept high, equal to 0.9, whereas the ϵ value gradually decreases from 0.9 (explorative strategy) to 0.1 (exploitative strategy). Later, in the convergence phase, the ϵ value is kept constant to 0.1, whereas l decreases from 0.9 to 0.1. Lastly, in the final phase, the values of l and ε are maintained small to rapidly adapt in case of changes in the environment.



(a) (b) (c) (d)

Fig. 3. Scenario with two (a), four (b), eight (c) and sixteen nodes (d).

TABLE I
PARAMETERS SETTING OF THE ATMEL ZIGBIT NODES

Parameter	Symbol	Value
Number of pairs nodes	n_n	1,2,4,8
Distance transmitter-receiver (m)	d	1
Distance between a pair of transmitters (m)	D	5
Packet payload size (byte)	$payload$	50
Inter-arrival time (ms)- Poisson Distributed	μ	300
Number of packets in a window	N	10
Number of transmission power levels	n_{ptx}	16
Transmission power (dBm)	p_{tx}	-17, -12, -8, -6, -4, -3, -2, -1, 0, 1, 2, 2.5, 3, 3.4, 3.7, 4
Number of states	n_s	68
Reward quantization step size	Δ	3.12
Reward quantization levels	m_r	320
PRR quantization levels	m_{prr}	20
Maximum retransmissions attempts	n_{retr}	3
Maximum CCA attempts	n_{cca}	4

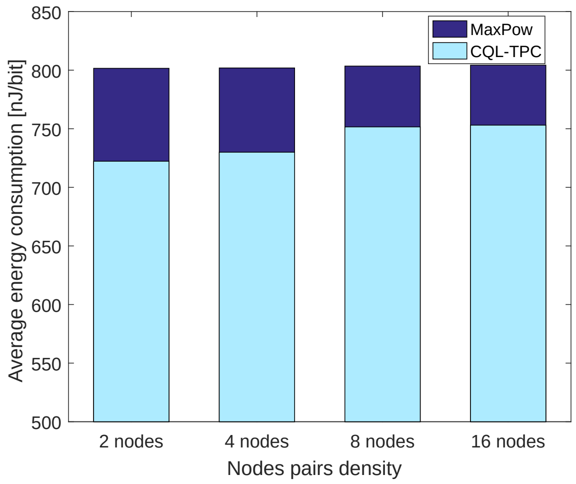
It is worth noting that, we assume a symmetric communication on the links between the nodes pairs to support the CQL-TPC scheme because each specific power level, computed by the cognitive algorithm, needs to be communicated by the transmitter to its recipient in the payload of the packets and, in the same way, the receiver uses the same power level to send back the acknowledgements.

B. Results and discussions

The performance of CQL-TPC is evaluated by comparing the proposed scheme with the standard strategy in which the

Performance metric	Nodes	CQL-TPC	MaxPow
PRR (%)	2	100	100
	4	100	100
	8	100	100
	16	99.22	99.85
Latency (ms)	2	4.2	4.7
	4	4.5	4.77
	8	4.7	4.87
	16	4.9	5.2
Retransmissions	2	0	0
	4	0	$1.7 \cdot 10^{-3}$
	8	$8.5 \cdot 10^{-3}$	$9.7 \cdot 10^{-3}$
	16	$15.6 \cdot 10^{-3}$	$20.5 \cdot 10^{-3}$
CCA attempts	2	1	0
	4	1	1.04
	8	1.04	1.08
	16	1.06	1.12
Convergence time (min)	2	4.7±0.5	-
	4	5.2±1.5	-
	8	11.1±2.6	-
	16	14.7±5.7	-

(a)



(b)

Fig. 4. Network performance comparison between CQL-TPC and MaxPow varying the density pairs: a) General network metrics; b) Average energy consumption.

TABLE II
PARAMETERS SETTING FOR THE PHASES OF LEARNING, CONVERGENCE AND TESTING

Phase	l	ε	δ	Episode #
Learning	0.9	0.9	0.8	$k < 150$
	0.9	0.7	0.8	$150 \leqslant k < 300$
	0.9	0.3	0.8	$300 \leqslant k < 450$
	0.9	0.1	0.8	$450 \leqslant k < 600$
Convergence	0.1	0.1	0.8	$600 \leqslant k < 750$
	0.01	0.1	0.8	$750 \leqslant k < 900$
Testing	0.01	0.01	0.8	$k \geqslant 900$

maximum power (MaxPow) is always used for the transmission. As performance indexes to validate the cognitive approach within the presented communication scenarios, we computed the average values of PRR, latency, number of retransmissions, number of CCA attempts and energy consumption of the whole network. We also evaluated the convergence time of the CQL-TPC scheme in relation to the density of the network. This is computed as the time needed by the agent to reach the Q-value that differs at most 1% from the final value, at the end of each experiment.

The obtained results are summarized in figure 4.a where it is possible to note that most of the performance between CQL-TPC and MaxPow are similar in all the scenarios. In particular, the PRR slightly drops when the network is composed by 8 pairs of nodes; on the contrary, the latency, the retransmissions and the CCA attempts raise proportionally to the number of nodes. The main and significant difference that confirms the value of the cognitive approach, is related to the energy consumption due to the used transmission power levels; in this case, the intelligence within the CQL-TPC allows to wisely adapt the transmission power saving more than 10% of the energy as shown in figure 4.b.

In particular, the difference in energy consumption between the two tested scheme varies between 27 nJ/bit and 15 nJ/bit, depending of the density of the network. Assuming to use two alkaline AA batteries of $3000mAh$ @$1.5V$ for each node corresponding to a total energy of 32400 Joule, the nodes in which the CQL-TPC is implemented can transmit for a longer period before batteries depletion. We have estimated that the lifetime of the nodes increases from several days (*i.e.*, 15-25), without any sleeping techniques, up to few months (*i.e.*, 9-16) if the deep sleep mode is enabled during the idle periods. Moreover, the energy consumed by the CQL-TPC in the learning and convergence phases does not affect the battery capacity in the long term since such estimated average value of about 2 Joule and 0.35 Joule, with deep sleep disabled and enabled, can be considered as a negligible value compared to the total batteries energy.

Finally, the convergence time of the proposed scheme is similar in the first two scenarios but it increases for denser networks in which the interference is higher and more dynamic; thus, the system is less stable and it requires more time to settle as confirmed by the high standard deviation values for the scenario with sixteen nodes. However, it is worth noting that the obtained converge time is not a big issue in static WSN scenarios in which the nodes slowly vary their mutual interference making the presented approach far preferable with respect to the deterministic one.

VI. CONCLUSION

Many powerful AI and ML techniques addressing knowledge engineering, planning, scheduling, and learning in distributed environments have been developed in last years to face different networking and communication issues taking advantages from the availability of both processing and storing capabilities within the Cloud. In the next future, the attention on the design and application of such powerful techniques will be moved from the *core* to the *edge* of complex and heterogeneous networks by adapting and re-designing their working features to face the limitation imposed by the presence of widely used constrained IoT devices.

In this perspective, the presented work discussed the potential and, at the same time, the needs of moving more *intelligence* towards the *edge* of the networks by making constrained devices evolve through the use of lightweight ML and AI techniques, paving the way for the so called *"actionable intelligence"* at the edge. The implemented case study, focused on the design of a cognitive transmission power control well suited for small real sensors, proves the feasibility of the proposal in terms of energy efficiency, reliability and scalability also ensuring standard network performance levels.

References

[1] M. Chen, V.C.M. Leung "From cloud-based communications to cognition-based communications: A computing perspective," *Computer Communications*, Vol. 128, pp. 74-79, 2018.

[2] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, F. Kojima "Big Data Analytics, Machine Learning, and Artificial Intelligence in Next-Generation Wireless Networks," *IEEE Access*, Vol. 6, pp. 32328-32338, 2018.

[3] P. Pace, G. Aloi, G. Caliciuri, G. Fortino "A Mission-Oriented Coordination Framework for Teams of Mobile Aerial and Terrestrial Smart Objects," *Mobile Networks and Applications*, Vol. 21, No. 4, pp. 708-725, 2016.

[4] Y. Luo, Y. Duan, W. Li, P. Pace, G. Fortino "Workshop Networks Integration Using Mobile Intelligence in Smart Factories," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 68-75, Feb. 2018.

[5] R. Li *et al.* "Intelligent 5G: When Cellular Networks Meet Artificial Intelligence," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175-183, October 2017.

[6] M. Chen and Y. Hao "Task Offloading for Mobile Edge Computing in Software Defined Ultra-Dense Network," *IEEE Journal on Selected Areas in Communications*, Vol. 36, no. 3, pp. 587-597, March 2018.

[7] H.H.W.J. Bosman, G. Iacca, A. Tejada, H.J. Wörtche, A. Liotta "Ensembles of incremental learners to detect anomalies in ad hoc sensor networks," *Ad Hoc Networks*, Vol. 35, pp. 14–36, Elsevier, 2015.

[8] D.C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, A. Liotta "Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science," *Nature Communications*, 9 (1), art. no. 2383, 2018.

[9] M. Chincoli, A. Liotta "Self-Learning Power Control in Wireless Sensor Networks," *Sensors*, Vol. 18(2), MDPI, 2018.

[10] K. Weiss, T.M. Khoshgoftaar, D. Wang, Q. Zhao, D. Grace, T. Clarke "A Survey of Transfer Learning," *J. Big Data*, Springer, 2016.

[11] Q. Zhao, D. Grace, T. Clarke "Transfer Learning and Cooperative Management: balancing the quality of service and information exchange in cognitive radio networks," *Trans. Emerging Tel.Tech.*, Vol. 26, pp.290-301. Wiley, 2014.

[12] M. Chen, U. Challita, W. Saad, C. Yin, M. Debbah "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," 2018. [Online]. Available: https://arxiv.org/abs/1710.02913

[13] H.H.W.J. Bosman, G. Iacca, A. Tejada, H.J. Wörtche, A. Liotta "Spatial anomaly detection in sensor networks using neighborhood," *Information Fusion Journal*, Vol.33, pp.41-56. Elsevier, 2017.

[14] D.C. Mocanu, E. Mocanu, P.H. Nguyen, M. Gibescu, A. Liotta "A Topological Insight into Restricted Boltzmann Machines," M*achine Learning Journal*, pp. 1-28. Springer, 2016.

[15] W. Qiang and Z. Zhongli "Reinforcement learning model, algorithms and its application," International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), Jilin, 2011, pp. 1143-1146.

[16] ZigBit wireless Modules AT86RF233 Datasheet. [Online] - Available at: http://www.microchip.com/design-centers/wireless-connectivity/embedded-wireless/802-15-4/hardware/zigbit-modules