Smart Measurement Systems Exploiting Adaptive LoRaWAN Under Power Consumption Constraints: a RL Approach

Tommaso Fedullo
Dept. of Management and Engineering
University of Padova, Italy and
Department of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
tommaso.fedullo@phd.unipd.it

Alberto Morato
National Research Council of Italy, CNR-IEIIT
alberto.morato@ieiit.cnr.it

Federico Tramarin
Department of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia
Modena, Italy
tramarin@unimore.it

Paolo Ferrari, Emiliano Sisinni
Dept. of Information Engineering
University of Brescia
Brescia, Italy
{paolo.ferrari,emiliano.sisinni}@unibs.it

2022 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT) | 978-1-6654-1093-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/METROIND4.0IOT54413.2022.9831487

***Abstract*—Nowadays, the development of smart, distributed and possibly wirelessly connected sensors networks is acquiring a greater importance. Indeed, the need to connect mobile or even unreachable devices (sensors, controllers and actuators) is mandatory in the novel smart factory context. Actually, the novel industrial measurement systems must guarantee certain levels of measurement accuracy, while suitably handling time-critical measurement and non - measurement data. Several commonly-used wireless standards can be employed, after careful protocol modifications, aiming at increasing both deterministic and real-time behaviour. Low Power Wide Area Networks (LPWANs) can be suitably adapted to cope with the stringent requirements of the industrial measurement scenario, due to the long communication range and low energy consumption. In particular, this paper focuses on one of them, LoRaWAN, that can be suitably configured through a set of communication parameters. For this reason, the paper aim is to propose and analyse a novel Reinforcement Learning - based adaptation strategy, to maximize the correctly received packets while lowering the energy consumption. The proposed adaptation policy has been compared with the Adaptive Data Rate (ADR) one, that is described by the standard. Results are encouraging, since the proposed methodology achieves better results in terms of correctly received packets compared to ADR, while maintaining similar levels of energy consumption.**

***Index Terms*—Reinforcement Learning, LoRa, LPWANs, ADR, Machine Learning, Artificial Intelligence**

I. Introduction

In the last years, the industrial panorama has been deeply modified, by the introduction of novel paradigms and technologies to accomplish the requirements coming from the novel smart factory vision. In this context, the development of IoT-based, smart and distributed measurement systems is of fundamental importance. Actually, increasingly performing communication networks must be exploited, aiming at the development of deterministic and robust measurement systems. The latter becomes remarkably critical in the industrial scenario, where noise and interference coming from the surrounding environment may deeply impact the communication performance, and in turn the measurement process. Several industrial test cases have underlined the need for wireless communication [1], to connect mobile and sometimes physically unreachable nodes. Physical access to nodes is critical in modern distributed measurement systems. In fact, energy consumption was not a concern in legacy industrial applications, leveraging on wired technologies. On the contrary, in applications where the sensors are not serviceable and are battery-powered, energy consumption is critical. In both of these scenarios, LPWANs, particularly LoRaWAN (Long Range) from Semtech, have been widely used due to their ability to manage a large number of nodes while maintaining an adequate data rate, transmission range, and energy consumption. Indeed, at the physical level (PHY), typically implemented by proprietary LoRa radios from Semtech, some parameters such as, beween others, Spreading Factor (SF) and Transmission Power (TP), can be changed to counteract channel impairments. LoRaWAN, in this regard, allows the use of algorithms for tuning these parameters, aiminig at increasing the Data Extraction Rate (DER) and generally improving the communication efficiency [2]. Moreover, by using a suitable adaptation strategy, it is possible to handle critical communications [3]. Among others, Adaptive Data Rate (ADR) is one of the most popular [4]–[6].

However, algorithms based on machine learning (ML), particularly those that use Reinforcement Learning (RL), have proven to be particularly effective. Indeed, in [7], the authors demonstrated how an RL-based rate adaptation algorithm outperformed ADR's performance but without taking power consumption into account. Nonetheless, RL's bases on a trial-and-error training activity, where a feedback from the environment is generated and used to choose the best actions, given the specific working conditions. In other words, the RL algorithm can be trained to maximize DER while maintaining a low power consumption and possibly outperforming ADR in this regard.

For these reasons, starting from the results in [7], we compare ADR and RL in terms of DER and power consumption. Moreover, we propose an enhanced RL algorithm capable of optimizing both of these metrics. The proposed approaches have been tested using simulations on simple but meaningful deployments.

The rest of the paper is organized as follows. Sec. II explain the theoretical foundations of LPWANs and Reinforcement Learning. Sec. III and IV describe the implemented architecture and simulation setup, respectively. The results are discussed in Sec. V. The paper is concluded in Sec. VI.

II. Theoretical Background

In this section a brief overview of the LoRaWAN specifications, the most diffused member of the LPWAN fmaily at the time of writing, is provided. Subsequently, the foundation of reinforcement learning is described.

A. LPWANs and LoRaWAN

LPWAN is a term grouping all the (wireless) communication technologies aiming at maximizing the coverage area tolerating low data rate and infrequent information exchanges. Due to the limited energy resource provided by battery supplied end devices, it is not possible to transmit over long distance simply increasing the transmission power; as a consequence, receivers have to be designed in order to maximize the sensitivity, which in turn it is possible if narrow bandwidth is considered. For this reason, LPWAN technologies all offer a raw bit rate in the range [0.1, 100]kbps. Nowadays, the most diffused example of LPWAN is the LoRaWAN solution, whose (open) specifications are drafted and managed by the LoRa Alliance. In particular, LoRaWAN standard aims at defining the upper layers of the communication protocols stack, while PHY services are usually implemented leveraging on proprietary LoRa radios (exploiting a solution developed and patented by Semtech). The LoRa physical (PHY) layer uses an enhanced version of the Chirp Spread Spectrum (CSS) modulation. Instead of using up- and down-chirp for encoding individual bits, as for the so called binary orthogonal keying (BOK), frequency trajectories are circularly wrapped over a fixed bandwidth B by SF discrete time steps. Consequently, the chirp duration T_C is divided into SF intervals and each chirp can encode SF bits. The actual chirp duration is chosen to satisfy the relationship $2^{SF} = T_C B$, so that frames transmitted using different SF values are quasi-orthogonal and can overlap in time and frequency (as if they were transmitted using different, virtual, channels). The use of large SF enhances the spreading gain, improving the signal-to-noise ratio (SNR), but the price to pay is a longer over-the-air duration. Operating in unlicensed bandwidths, LoRa must respect local norms depending on the adopted frequency plan; for instance, when used in Europe, the EU868 is typically considered, exploiting the available band in the 868MHz region. As a consequence, the transmission power TP and the duty-cycle are both limited. The Data Link Layer (DLL) includes pure ALOHA as the medium access strategy, which can be complemented by the so-called Channel Activity Detection (CAD), to permit to Listen Before Talk (LBT). The network (NWK) topology mimics the mobile communications and it is based on the "hub-and-spoke" idea, in which end devices are single hops away from a gateway (GW). The gateway(s) have wired connections towards the backend, which is typically implemented in the cloud and can leverage on large computational and storage capabilities. End devices trigger data exchanges, sending a frame in the uplink direction (towards the GW), and subsequenlty opening two reception windows (RX1 and RX2), for downlink frames ($TRX1$ and $TRX2 = TRX1 + 1s$ after the end of the uplink frame; $TRX1, 2$ usually equals [1,2]s for data transactions, whereas $TRX1, 2$ equals [5,6]s for network joining). Interesting to note, the LoRaWAN specifications also detail the backend architecture, defining numerous logical entities, known as the Network Server (NS), the Application Server (AS), and the Join Server (JS). The NS is in charge of managing the network and implements Adaptive Data Rate (ADR) strategies. The ADR algorithm described in the standard is responsible for deciding the SF and TP values depending on the uplink SNR margin $SNR_M = SNR_{Meas} - SNR_{DR} - M$, being SNR_{Meas} the maximum measured SNR for the last 20 received uplink frames, while SNR_{DR} is the threshold for ensuing correct frame decoding and $M = 10dB$. The chosen configuration is provided to the end device by means of the aforementioned downlink frames; if connectivity is lost, the mote autonomously operates increasing the TP and the SF until the bidirectional communication is regained. Two configurable thresholds regulates the ADR behavior on the end device, namely ADR_ACK_LIMIT (typically set to 64) and ADR_ACK_DELAY (typically set to 32). If ADR_ACK_LIMIT uplink frames have been transmitted without any downlink frame reception, an explicit ADR_ACK downlink frame is requested. If such a frame is not received in the successive ADR_ACK_DELAY receiving opportunities, the end node starts increasing the TP first, and then the SF. The NS also performs message integrity checks, leveraging on network wide session key. The AS is designed to facilitate end user integration and takes care of application payload (de)ciphering, using a different application session key. These session keys are obtained during the provisioning, that generally occurs according to the OTAA (Over The Air Activation) procedure. Since the LoRaWAN 1.X release, the JS has been introduced, managing the keys and permitting a more secure decoupling between data management and network management.

B. Reinforcement Learning

Among all the different Machine Learning (ML) techniques, Reinforcement Learning (RL), at present, is one of the most attractive, with several application fields in which it was implemented [8]. In particular, the modelization of a RL policy is depicted in Fig. 1.

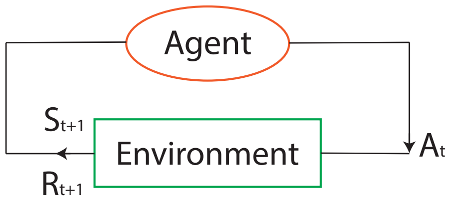

Fig. 1. Reinforcement Learning (RL) components

The environment (i.e. the system) is modeled through the definition of suitable states, being the states set named S. The RL policy foresees to train an Agent, to make it learn how to properly act on the environment. In particular, a set of actions A must be defined, being each action $A_i \in A$ a possible choice to allow the transition, between states. In particular, at the time interval t, the chosen action A_t enact the transition between the initial state S_t and the final state S_{t+1}. The training phase of the algorithm is carried out by using a trial and error methodology, where the goodness of each chosen action is evaluated through the definition of a reward function, r. Summarizing, each RL technique foresees the definition of a suitable Markov Decision Process ($MDP = \{S, A, P, r\}$), where P is the probability function. At each training step t the agent chooses an action, causing the transition between the state S_t and S_{t+1}. Actually, a reward R_{t+1} is produced (describing the validity of the performed action A_t) by evaluating the reward function, i.e. $R_{t+1} = r(t+1)$. Specifically, the aim is to fill a table where at each pair $\{(S_i, A_i) \mid S_i \in S, \quad A_i \in A\}$ is associated a suitable rate, evaluating the goodness of the action A_i given the state S_i. Afterwards, during execution, the so-called best policy π is exploited, by using (at each state S_t) the best evaluated action A_t. Usually, the function $V^\pi(s) = E_\pi\{R_t|S_t = s\}$ or the function $Q^\pi(s, a) = E_\pi\{R_t|S_t = s, A_t = a\}$ can be employed to evaluate the cumulative rewards associated with the specific best policy π, named respectively value and action-value function. In the formulas above, when performing the policy π, E_π represents the expected return value. Actually, the evaluation of either the $V^\pi(s)$ or $Q^\pi(s, a)$ functions, can be carried out by using several different approaches. In this article, the Q-value associated to a specific pair (S_t, A_t), is calculated using Eq. (1).

$$\begin{aligned} Q(S_t, A_t) = Q(S_t, A_t) + \\ + \alpha * [R + \gamma * Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)] \end{aligned} \tag{1}$$

Specifically, Q-values are calculated by using $S_t, A_t, R, S_{t+1}, A_{t+1}$, namely State Action Reward State Action (SARSA) algorithm. Actually, during training, the Q-values are updated, while the best ones are used when exploiting π. In Eq. 1 the first step is to calculate the so-called *target*, $T = R + \gamma * Q(S_{t+1}, A_{t+1})$. In particular $\gamma \in \{0, \ldots, 1\}$ is chosen to suitably weight the importance given to the expected Q-value. Secondly, the learning rate α is used to properly weight the error between T and the previous Q-value. In particular, the higher α the higher the possibility to modify the Q-values. Usually, α and γ are set after a careful analysis of the system and a trial and error phase. Finally, during training, an $\epsilon - greedy$ algorithm (Eq. 2) is used to evaluate a trade-off between exploration and exploitation.

$$\begin{cases} A_{t+1} = rand() \quad if \quad n \geq \epsilon \\ A_{t+1} = max(Q(S_t)) \quad if \quad n < \epsilon \end{cases} \tag{2}$$

In particular, exploration foresees to test a randomly chosen action, while exploitation foresees to use the actual best action, to assure that all actions are going to be evaluated during training. Referring to Eq. (2), $n \in [0, 1]$ represents a real random number, and ϵ can be suitably modified to achieve the desired trade-off between exploration and exploitation.

III. Reinforcement Learning - based Adaptive LoRaWAN

The development of performing and accurate industrial IoT measurement systems, poses several challenges in terms of correctly delivered packets, under strict timing requirements. In this context, LoRaWAN networks can be effectively implemented, due to their performance in terms of correctly delivered packets, transmission range and cost. Moreover, LoRa radios networks offer the possibility to effectively configure the communication parameters, such as SF and TP. This feature is extremely important in the industrial scenario, being the latter particularly adverse in terms, for example, of noise and coexistence with other devices. Moreover, environmental conditions may change a lot, thus underlining the importance of the development of ever performing techniques devoted to communication parameter's adaptation. In this context, the Data Extraction Rate (DER), defined as per Eq. 3, acquires a fundamental importance.

$$DER_t = \frac{RP_t}{SP_t} \cdot 100 \quad (\%) \tag{3}$$

Actually, SP_t and RP_t are, respectively, the sent and the correctly received packets before the specific instant of time t. Moving from the aforementioned observations, in a previous work [7] we proposed an RL–based Adaptation Algorithm (RLAA). The latter, bases its behaviour on a suitable MDP, where states, actions and rewards are defined as follows. First of all, states are defined as couples of DER and SNR values. In particular, both DER and SNR variation intervals of the UpLink messages are discretized to suitably describe all the possible communication conditions. Ten different equally spaced DER levels have been identified, ranging from 0% to 100%. Moreover, after a careful analysis of the behaviour of the simulator, 12 SNR levels ranging between 0 and $60dB$, and one level to represent $SNR > 60dB$, have been recognized. The proposed discretization allows to describe the system environment by means of 130 different states. The SNR becomes, in this context, a key parameter, able to suitably represent the channel's condition, depending on both the distance between node and gateway and the interference with other devices. For

TABLE I
COMMUNICATION PARAMETERS (SF, TP) POSSIBLE VALUES (EU 868).

Name	Possible values
SF	7, 8, 9, 10, 11, 12
TP	2, 5, 8, 11, 14

this reason, the Q table resulting from the training activity can be shared between all nodes, thus developing a smart, mobile and plug and play measurement system. Secondly, Actions are defined as couples of TP and SF, whose possible values are presented in Table I, thus defining 30 different actions.

After a trial and error phase, rewards for RRAA have been defined as per Eq. (4), to suitably optimize the total DER.

$$r = \frac{(DER_t - DER_{t-1}) * SNR_t}{SF_t} \tag{4}$$

Specifically, higher rewards are given to actions resulting in higher DERs, SNRs and lower SFs, i.e. higher data rates. The latter, in particular, aims at accomplish a trade–off between communication robustness and low transmission delay. This adaptive strategy does not take into account the energy consumption, being suitable for applications where the battery lifetime of the sensors is not critical. On the contrary, there are several industrial applications where measurement systems must also accomplish to battery lifetime requirements. For example, in our previous article [9], we presented an additive manufacturing test case where sensors are embedded into the final product. In this context, sensors become unreachable, thus underlining the importance of the battery lifetime. For these reasons, in this paper a modification of the RL rewards is presented, together with a deep analysis of the energy consumption of each adaptive strategy. The novel RL algorithm, namely Green RRAA (GRRAA), foresees to modify the rewards, to take into account the TP. The new reward function is presented in Eq. 5.

$$r = \frac{(DER_t - DER_{t-1}) * SNR_t}{\beta * SF_t * TP_t} \tag{5}$$

It is worth observing that, the β value can be used to suitably tune the RL adaptation policy. In particular, the higher β, the higher the importance of SF and TP in the rewards. In this situation, much attention is given to the energy consumption, as the algorithm tries to use lower TPs and SFs. Actually, the lower the SF, the higher the data rate, the lower the transmission time, thus reflecting in a lower energy consumption and transmission delay. On the contrary, the lower β, the higher the importance of the DER. In conclusion, α and γ values of Eq. 1 are set, respectively, to 0.1 and 0.7, while β is set to 10.

IV. SIMULATION SET-UP

In this paper, LoRaEnergySim (LES) [10], a Python-based simulation framework, has been used to properly evaluate the behaviour of the three algorithms (RRAA, GRRAA and ADR). In particular, LES allows to suitably simulate collided and weak packets. Moreover, as the simulator already implements the ADR adaptive technique, it is particularly suitable to compare ADR with the proposed adaptive strategy in terms of DER. In addition, an accurate energy consumption model is used by LES, thus allowing to achieve meaningful results in terms of used power.

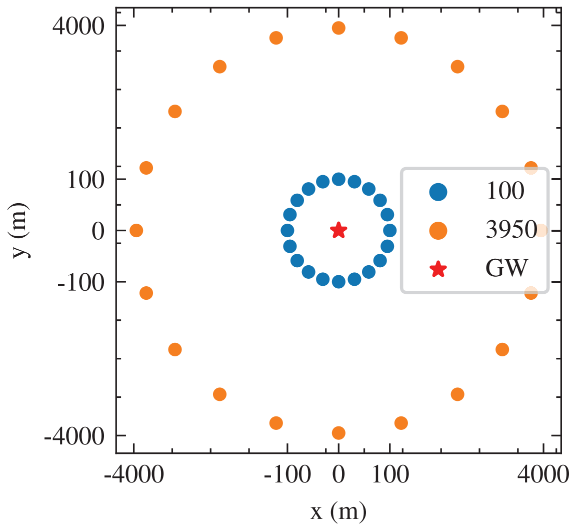

Fig. 2. Locations of the nodes

The LES provides objects for modeling one or more "Nodes", a single "Gateway" and a suitable "Air Interface". The latter one suitably models the propagation bahavior, the SNR, and the collisions. In this way, the communication between several Class-A LoRa nodes has been suitably simulated. As a first step, two simulation scenarios have been considered. As can be seen in Fig. 2, nodes are placed in circumferences centered on the gateway, respectively with a radius of 100 m and 3950 m. In both scenarios, a total of 20 nodes has been used. This particular choice has been made to obtain simple but meaningful environments allowing a clear analysis and comparison of the transmission parameters selected by the three algorithms, depending only by the distance between the node and GW. In the depicted scenarios, the network backend (BE), after the reception of N_{step} consequent uplink messages, sends a downlink message communicating the RP_t value of the specific node. The latter information allows the DER calculation as per Eq. (3).

Table II resumes the simulated environments setup.

TABLE II
SIMULATION SETUP

Parameter	**Value**
Number of nodes	20
Area of the simulated environment	64 km^2
Total UpLink packets per node (N)	1000
Period of confirmation downlink message (N_{step})	5
Payload length ($Byte$)	50
Duty Cycle (%)	1

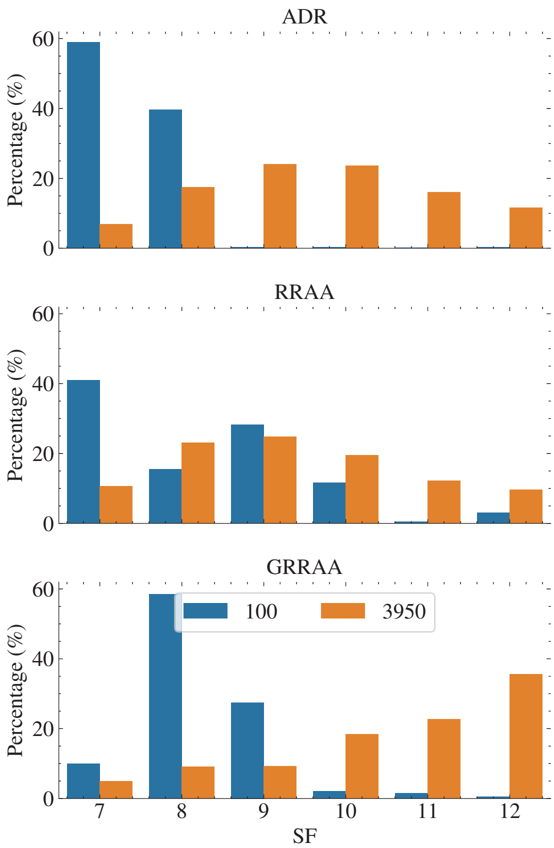

Fig. 3. Comparison of the SF in two different scenarios with ADR, RRAA, and GRRAA

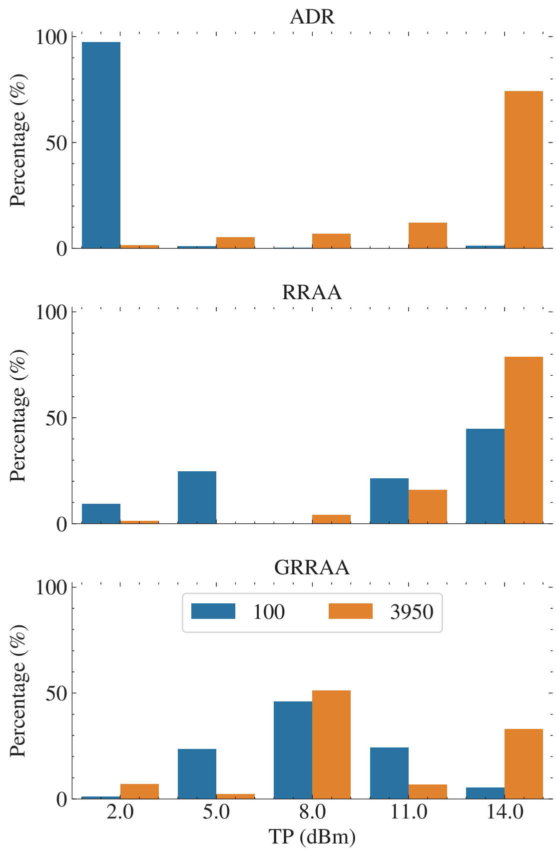

Fig. 4. Comparison of the TP in two different scenarios with ADR, RRAA, and GRRAA

It is worth noting that, in Table II, the duty cycle is set to 1%. Then, the inter-message delay has been set to the highest value, to suitably respect the imposed duty cycle also when the data rate is the lowest (i.e. SF12). In the next section, simulation results are presented and discussed.

V. Results and discussion

After an extensive training phase, it was possible to test the adaptation algorithms, in all the aforementioned scenarios. In particular, the chosen SF and TP values are presented respectively in Fig. 3 and in Fig. 4.

The most important observations can be derived from Fig. 4. In particular, the RRAA algorithm does not optimize the choice of TP, that has been mostly set at the higher values. Clearly, this involves in higher DER compared to ADR, that optimises also the energy consumption. On the contrary, the GRRAA better optimises the usage of TP. By comparing the behaviour of GRRAA and ADR, it is possible to underline that the network presents different behaviours in the two considered scenarios. Indeed, ADR chooses mainly the lower TP when the distance from the gateway is very low, while setting high TPs when dealing with more challenging scenarios. Actually, GRRAA balances better the TPs in both scenarios. By analysing the SF, it is also possible to underline that, by highering the β value in Eq. 5 it is still possible to lower the energy consumption, and also the transmission time. Indeed, GRRAA presents higher DERs because it makes use of higher SFs (especially when the distance from the gateway is high), compared to ADR. This leds to higher transmission times and energy consumption. Moreover, comparisons between the three algorithms in terms of DER and Power spent per Packet (PP) are depicted respectively in Fig. 5 and 6. Furthermore, the mean DER and PP values are listed in Table III.

TABLE III
Comparison of the Power per Packet in two different scenarios with ADR, RRAA, and GRRAA

Name	SimNo	DER	PWR per PKT	PWR over DER
ADR	100	99.88	182.08	1.82
ADR	3950	73.25	233.33	3.19
RRAA	100	100.00	216.65	2.17
RRAA	3950	83.12	236.69	2.85
GRRAA	100	99.77	199.37	2.00
GRRAA	3950	76.60	215.12	2.81

Results in terms of both DER and PP, are totally consistent with the analysis made above. When the distance from the gateway is too low, all the algorithms are capable to correctly deliver packets, and ADR performs better in terms of energy consumption. On the other side, GRRAA performs definitely better in terms of PP, while maintaining high DERs, when the distance from the gateway becomes higher. From Table III, it is possible to draw some last observations. Firstly,

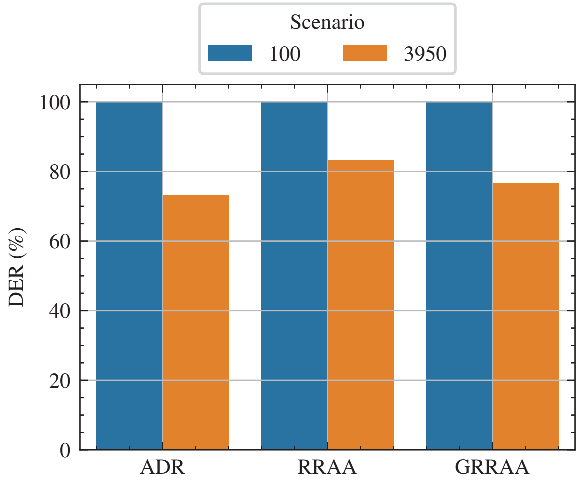

Fig. 5. Comparison of the DER in two different scenarios with ADR, RRAA, and GRRAA

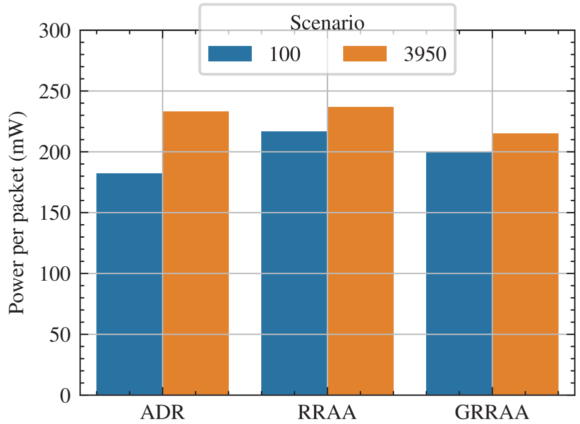

Fig. 6. Comparison of the Power per Packet in two different scenarios with ADR, RRAA, and GRRAA

all the algorithms perform better in terms of DER when the distance from the GW is lower, thus underlining the correctness of the proposed algorithms. Moreover, the power spent to send a single packet grows with the distance from the GW. Summarizing, both the simulation setup and the algorithms implementation are coherent with the theoretical behaviour of the network, and the proposed approach is solid and performs better than ADR. Finally, we also considered a scenario where 60 nodes are randomly placed in a $1km^2$ area, with a central gateway. The results obtained in such scenario, are presented in Table IV.

Actually, the results are encouraging as they confirm the previous ones, allowing to properly generalise the discussion made above.

TABLE IV
COMPARISON OF THE POWER PER PACKET ON THE RANDOM LOCATIONS SCENARIO WITH ADR, RRAA, AND GRRAA

Name	DER	PWR per PKT	PWR over DER
ADR	94.04	192.13	2.04
RRAA	98.94	210.75	2.13
GRRAA	98.97	198.19	2.00

VI. Conclusions and Future directions of Research

In this paper, a novel RL adaptation technique for LoRaWAN networks has been proposed. Moreover, particular attention was given to the energy consumption, thus focusing on wireless and mobile industrial measurement systems. In this context, particular attention must be given to the number of correctly received packets, as it impacts on the quality of the measurement process. Actually, a novel RL adaptation strategy has been proposed (GRRAA) and a properly tuned and widespread Python based simulation framework has been exploited to realistically test GRRAA, comparing it with the widespread ADR strategy. From the obtained results, the proposed algorithm allows to increase the DER, while maintaining the same energy consumption of ADR. Moreover, the RL strategy can be properly tuned to choose the best trade off between energy consumption and DER, for the specific field of application. Actually, we expect that this work will become a baseline for the following implementation in a real-world scenario. Indeed, simulations were mandatory as, in order to be able to fill all the Q-Table during exploration, a wide network was needed, that is hardly to develop due to complexity and cost reasons. In light of the encouraging realistic results obtained, in future works we plan to make an experimental sensor network incrementally learn, starting from the obtained Q-Table, how to adapt LoRa parameters.

References

[1] Y. Liu, M. Kashef, K. B. Lee, L. Benmohamed, and R. Candell, "Wireless network design for emerging iiot applications: Reference framework and use cases," *Proceedings of the IEEE*, vol. 107, no. 6, pp. 1166–1192, 2019.

[2] L. Beltramelli, A. Mahmood, P. Österberg, M. Gidlund, P. Ferrari, and E. Sisinni, "Energy efficiency of slotted lorawan communication with out-of-band synchronization," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–11, 2021.

[3] E. Sisinni, D. F. Carvalho, and P. Ferrari, "Emergency communication in iot scenarios by means of a transparent lorawan enhancement," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10 684–10 694, 2020.

[4] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "A Survey on Adaptive Data Rate Optimization in LoRaWAN: Recent Solutions and Major Challenges," *Sensors*, vol. 20, no. 18, p. 5044, Sep. 2020.

[5] V. Moysiadis, T. Lagkas, V. Argyriou, A. Sarigiannidis, I. D. Moscholios, and P. Sarigiannidis, "Extending adr mechanism for lora enabled mobile end-devices," *Simulation Modelling Practice and Theory*, vol. 113, p. 102388, 2021.

[6] K. Anwar, T. Rahman, A. Zeb, I. Khan, M. Zareei, and C. Vargas-Rosales, "Rm-adr: Resource management adaptive data rate for mobile application in lorawan," *Sensors*, vol. 21, no. 23, p. 7980, 2021.

[7] T. Fedullo, A. Morato, F. Tramarin, P. Bellagente, P. Ferrari, and E. Sisinni, "Adaptive lorawan transmission exploiting reinforcement learning: the industrial case," in *2021 IEEE International Workshop on Metrology for Industry 4.0 IoT (MetroInd4.0 IoT)*, 2021, pp. 671–676.

[8] Y. Li, "Reinforcement learning applications," 2019.

[9] L. Trevisan, S. Vitturi, F. Tramarin, and A. Morato, "An iiot system to monitor 3d–printed artifacts via lorawan embedded sensors," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 1331–1334.

[10] G. Callebaut, G. Ottoy, and L. van der Perre, "Cross-layer framework and optimization for efficient use of the energy budget of iot nodes," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–6.