Lightweight IDS For UAV Networks: A Periodic Deep Reinforcement Learning-based Approach

Omar Bouhamed[1], Ouns Bouachir[2], Moayad Aloqaily[3], Ismaeel Al Ridhawi[4]

[1]Concordia University, Montreal, QC, Canada, omar.bouhamed@outlook.com
[2]College of Technological Innovation, Zayed University, UAE, ouns.bouachir@zu.ac.ae
[2]Faculty of Engineering, Al Ain University, UAE, maloqaily@ieee.org
[4]Kuwait College of Science and Technology, Kuwait, i.alridhawi@kcst.edu.kw

***Abstract*—The use of intrusion detection systems (IDS) has become crucial for modern networks. To ensure the targeted performance of such networks, diverse techniques were introduced to enhance system reliability. Many network designs have adapted the use of Unmanned Aerial Vehicles (UAVs) to provide wider coverage and meet performance targets. However, the cybersecurity aspect of UAVs has not been fully considered. In this paper, we propose a lightweight intrusion detection and prevention system (IDPS) module for UAVs. The IDPS module is trained using Deep Reinforcement Learning (DRL), specifically Deep Q-learning (DQN), to enable UAVs to autonomously detect suspicious activities and to take necessary action to ensure the security of the network. A customized reward function is used to take into consideration the dataset unbalanced nature, which encourages the IDPS module to detect minor classes. Also, considering the limited availability of resources for UAVs, a periodic offline-learning approach is introduced to ensure that UAVs are capable to learn and adapt to the evolution of intrusion attacks autonomously. Numerical simulations show the efficiency of the proposed IDPS in detecting suspicious activities and corroborating the advantages brought by the periodic offline learning in comparison with similar online learning approaches, in terms of accuracy and energy consumption.**

***Index Terms*—DRL, intrusion detection and prevention system, periodic offline learning, lightweight models, UAVs.**

I. Introduction

Over the past few years, the use of UAVs, also known as drones, has been proven to be very effective in supporting Internet of Things (IoT) systems. These flying IoT devices play a prominent part in the newly introduced communication and networking architecture to address the exponential growth of traffic demand from other fixed and mobile IoT devices [1], [2]. UAVs equipped with communication transceivers can be used to serve as mobile IoT Access Points (AP) to transmit, broadcast, capture, and gather information, thanks to their versatility, mobility, as well as their capability to supply line-of-sight (LoS) communication channels [3], [4].

UAVs are employed to a diverse set of environments, which some may be considered as either hostile and insecure, from a networking perspective, and thus, are exposed to different types of attacks. As such, there is a need to ensure the security of these units, to guarantee reliable and trustworthy communication of information between the various network components against intruder attacks [2]. To overcome such obstacles, intrusion detection and prevention systems (IDPS) were introduced to enable autonomous network and traffic monitoring to detect signs of a possible attack and prevent exploiting network vulnerability [5]. Such exposure usually comes in the form of malevolent data sent by the attackers to interrupt the UAV operation and even gain control over it, to eventually access all the rights and permissions available to the compromised unit [6]. When detecting a potentially malicious activity, the IDPS intervenes to stop the attack, either by dropping the dangerous packets, blocking the attacker's access, or resetting connections, in addition to sending an alert to the central station.

In this work, we propose a framework that empowers UAVs with the necessary intelligence to self-determine the security of its activities and to act accordingly when security threats are detected. The solution implements an efficient and sophisticated Reinforcement Learning (RL)-based module on each flying unit. Furthermore, the Artificial Intelligence (AI) module is coupled with feature selection methods that help select pertinent features, to decrease the needed computational power and improve detection accuracy. To support autonomy, a periodic offline-learning method is proposed to grant UAVs the capability to learn and adapt to changes in attack patterns.

Through data exchange, relayed by the fleet of UAVs to the central station via cloud servers, a global model is trained in the station to ensure the IDPS is active. In contrast to the online-learning approach, where each UAV updates its IDPS model based only on the received data, the proposed periodic offline-learning method uses the entire recently exchanged data by the fleet of UAVs, to build a more robust and efficient IDPS model. As the central unit receives new data, it keeps updating the global IDPS accordingly. Once a UAV returns to the docking station, the IDPS, which is implemented on the flying unit, is automatically updated by overwriting the existing, local version by the global AI-module trained in the central station. As a result, the framework ensures the security of the network in general, and the UAVs in particular, by enforcing the system with a smart first layer of defense, constantly updated to face any emerging novel attack.

This study concludes with a comparative analysis with the classical ML techniques and the online-Learning method to

978-3-903176-32-4

validate the performance of the periodic DRL-based approach, in terms of accuracy and energy consumption. The rest of the paper is organized as follows: Section II discusses some related work in the literature. Section III presents briefly a global overview of the proposed system, as well as, the techniques used to manage the used dataset. Section IV presents the proposed solution. The section includes an explanation of the proposed DRL-based classifier and the periodic offline-approach. Section V discusses the evaluation of the established methods and simulation results. Finally, a conclusion is drawn in Section VI.

II. Related Work

Only recently, a few works fully considered the cyber-security aspect of flying IoT devices. For instance, in [7], the authors dedicated their work to identify malicious activities over UAV sensors, by gathering the necessary information from statistical analysis techniques and the physical system. The presented work addresses the problem of identifying sensor attacks on UAVs and considers only spoofing attacks. In [8], the authors designed a hierarchical intrusion detection method responsible for monitoring UAV behavior, categorizing its comportment following a set of predefined rules, with the purpose of identifying suspicious activities (mainly GPS jamming, spoofing, and gray hole attacks). In [9], the authors, with the main focus on signal spoofing and jamming attacks, proposed a deep learning-based IDS for UAVs to identify intruders and ensure the safe return of these gadgets in case of loss of connection with the central unit. The authors used Self-Taught Learning along with a multi-class classifier, namely Support Vector Machine (SVM), to design the presented intrusion detection system.

In contrast, numerous traditional Machine Learning (ML) methods were studied for anomaly detection in IoT devices. In [10], the authors proposed an intrusion detection model based on a ML technique, namely, the Decision Tree model. They adopted a centralized architecture, where the system is enforced by IDS, implemented on the IoT Gateways (i.e a central unit where all data is gathered and processed). In [11], the paper presented an SVM-based IDS for anomaly detection. The proposed framework, based on analysis made on the packet arrival rate to the node, detects and classifies any malicious activities. In [12], the authors proposed an anomaly detection mechanism based on a game theory, where they used Nash equilibrium, to activate the anomaly detection framework to discover unfamiliar attack patterns when needed. In [13], the authors, using ML techniques, proposed enhancing the IoT system security by placing AI-based IDS modules at the system gateway level, with the objective of detecting and classifying suspicious activities.

Even though the discussed solutions provided good techniques to detect and classify anomalies in the system, their limitation lies in the adaptability to changes in the attack patterns and their suitability for energy and computationally-constrained devices, such as UAVs. As IDPS systems are becoming more sophisticated, attacks with new patterns emerge, which introduces more complication in detecting malicious attacks. To overcome such a dilemma, the existing solutions require frequent updates by re-training and revising the entire learning process with a new diverse dataset. Considering the frequent human intervention, the resources involved, and the immense computational cost, such process is neither efficient nor practical. Another major point to take into consideration is that traditional IDPS systems are usually centralized, where the network traffic analysis is conducted at a security central unit to determine the security of the network. IoT devices in general may lack the intelligence, but in our context, UAVs, having enough computational power, are capable of making decisions without relying on a central center [14]. However, this brings us to one of the crucial issues related to UAVs, their limited battery capacities [15]. As UAVs are energy-restricted, the design of lightweight IDPS mechanisms is exceedingly difficult, however, such systems are necessary to achieve and maintain UAV network security and act as a first layer of defense.

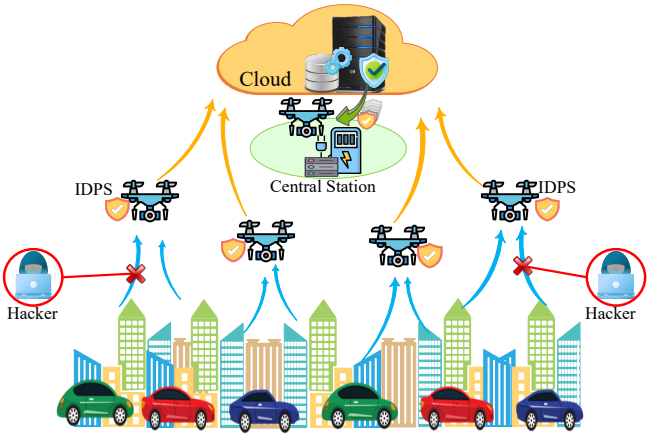

Fig. 1: System overview.

III. System Overview and Data Preparation

In this study, we propose a cost-effective IDPS through a distributed architecture, where a swarm of UAVs is employed to cover the maximum possible number of clients or events as depicted in Fig. 1. The chosen system design is generic and can be applied for a wide variety of applications such as search and rescue missions, delivering parcels, and traffic monitoring.

A. System Overview

One of the main objectives in this work is to displace a first line of defense, lightweight IDPS, UAV-aided network. An ML-based IDPS module is implemented on each UAV to ensure that each UAV possesses enough intelligence to detect intrusions and malicious activities. It does so by observing its present behavior and comparing it with acceptable activity patterns. Furthermore, to ensure that the AI-model is up to date, in the central station, the recently relayed data by the fleet of UAVs is continuously used to update a global IDPS. We assume that UAVs update their IDPS module

once they return to the central station to recharge. Moreover, the proposed solution ensures real-time detection. Such early sensing will ensure system robustness against attack attempts on the main network to obtain access to UAV applications and clients' data. In general, the IDPS system has two use cases:

- In case of normal behavior, the IDPS grants the client a safe passage, namely, a safe connection to exchange data.
- In case of an attack, a security protocol is triggered, where the IDPS blocks the client, in this case, the attacker finishes the session and drops the malicious packets. An alert is then sent to the security central station, in case of the need for further actions.

B. Dataset preparation and Feature selection

Due to the ever increasing cases in the number of cyber attacks in the last decade, many IDPS have been proposed. To evaluate the performance of such systems, the Canadian Institute of Cybersecurity presented a state of art dataset named CICIDS2017 [16]. In this paper, CICIDS2017 is chosen as it covers the latest threats which were not addressed by older datasets. The dataset is well-organized and cleansed and contains imbalanced data. It consists of 3,119,345 data samples and 83 features. The dataset instances are introduced as 15 different classes, labeled as normal activities and 14 types of malicious attacks, which provides a good reflection of real-world traffic.

Generally, numerical datasets have missing values, also known as Null values, which may affect the efficiency of the ML model. Hence, data cleaning and processing techniques are applied to deal with such a problem. In this paper, we choose to drop columns with more than a fixed threshold of missing values, while replacing the null values with either mean, mode, or median in the remaining columns. Additionally, features with unique values are dropped, since such features have no effect on the learning process of the model. As a result, the total number of features was reduced from 83 to 70. After cleaning the data, feature scaling was applied, which is considered a necessity for ML methods to avoid the possibility that some features with large values may have significant influence on the final output of the ML-model. To this end, the features were scaled by calculating its z-score, redistributing the data in such a way that its $mean = 0$ and its $standarddeviation = 1$.

Feature Selection is one of the core concepts in ML which has a huge impact on the performance of an ML-model. For these purposes, a filter-based method was used, namely Pearson Correlation, to keep only the significant features, with the lesser correlation having the target output. In this study, one of our main objectives is to ensure that the proposed IDPS model is lightweight. The number of used features plays a significant role, hence it is preferable to have fewer numbers of used features. However, the affect of lowering the number of features on the accuracy of the model should be considered. After repeatedly applying this process and training the model with a different number of features, we concluded that the optimal choice for ensuring model accuracy and cost-efficiency is 10 features. Therefore, the number of features was reduced from 70 to 10.

IV. Proposed Solution

As mentioned previously, the proposed framework is divided into two main processes, namely, applying an RL module on the UAVs, and implementing an offline periodic learning that is adapted to the central station.

A. RL for Classification

A well-known DRL algorithm was adapted in this work, namely DQN, to build a multi-class classifier to detect intrusions using a labeled dataset. Since classical RL agents learn from a live interaction with its surrounding environment, few changes were made. The environment was replaced with a sampling function of stored normal/intrusion cases. In addition to sampling the training dataset, the new pseudo-environment generates rewards based on the decision made by the model. In other words, the environment transform the data into a series of step-by-step decision problem. As the training set is presented in a list of 2-tuples (x_k, l_k), where x_k denotes the k^{th} sample and l_k its label, the DQN algorithm is executed for M episodes, such that each lasts T steps (as explained in Algorithm-1), where at each step t a random sample of the training set is selected. The selected x sample serves as the current state of the RL agent s_t. Label l is used as a parameter to evaluate the agent decision, action a_t.

Algorithm 1: DQN Training process

P_e = Init_Pseudo_Environment().
B = Init_Replay_Memory(N). // N: the capacity
$\mathcal{Q}$ = Init_Network(θ).// $\mathcal{Q}$: action-value network, θ: random weights
$\hat{\mathcal{Q}}$ = Init_Network($\hat{\theta}$). // $\hat{\mathcal{Q}}$: target network, $\hat{\theta} = \theta$
for *episode = 1,...,M* **do**
  Receive first observation $s_1 = x_1$.
  for t = *1,...,T* **do**
    if $\epsilon > random()$ **then**
      a_t = select_random_action($\mathcal{A}$)
    else
      a_t = argmax$_a \mathcal{Q}(s_t, a_t)$
    r_t, s_{t+1}= Execute_Action(a_t, P_e)
    $B \Leftarrow$ Store_Transition($[s_t, a_t, r_t, s_{t+1}]$).
    $[s_j, a_j, r_j, s_{j+1}]$ = Sample_Random_transitions(B).
    Set $y_j = r_j + \gamma \max_{a'} \hat{\mathcal{Q}}(s_{j+1}, a')|\hat{\theta})$.
    Perform_GradientDescent($(y_j - Q(s_j, a_j|\theta))^2$).
    if $k \pmod t \equiv 0$ **then**
      Update_TargetNetwork($\hat{\mathcal{Q}}$, $\mathcal{Q}$, k).
      // k: update frequency

We adopted a problem of multi-class classifications where the dataset was divided into a normal activity class and several anomaly classes. Two different rewards functions were tested, namely, a simple reward function and a customized reward function. In the first, the pseudo-environment generates 1 in case of a correct decision, action, and 0 in case it was wrong. Therefore, the action space for our agent is presented as $\mathcal{A} = \{0, 1, \ldots, n-1\}$, where n is the number of investigated classes and the reward is calculated as follows:

$$r_1(s_t, a_t, l_t) = \begin{cases} 1, & \text{if } a_t = l_t, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

In the second reward function, the pseudo-environment encourages the agent to detect minor classes, by granting it a high reward in case of a correct decision and a higher penalty otherwise. Hence, the action space remains equal to $\mathcal{A} = \{0, 1, \ldots, n-1\}$. The reward function is modeled as follows:

$$r_2(s_t, a_t, l_t) = \begin{cases} +\beta, & \text{if } a_t = l_t \text{ and } s_t \in \mathcal{C}_m, \\ +\zeta, & \text{if } a_t = l_t \text{ and } s_t \in \mathcal{C}_M, \\ -\zeta, & \text{if } a_t \neq l_t \text{ and } s_t \in \mathcal{C}_M, \\ -\beta, & \text{if } a_t \neq l_t \text{ and } s_t \in \mathcal{C}_m, \end{cases} \quad (2)$$

where $1 \geq \beta > \zeta > 0$, $\mathcal{C}_M$ and $\mathcal{C}_m$ are the majority classes and the minority classes, respectively.

B. Periodic Offline-Learning

The offline trained IDPS module should be efficient once it is implemented in the system. However, over time, with the increasing number of cyber-attacks and its non-stop evolution, updating the IDPS module policy is necessary. In this context, a periodic offline-learning approach was proposed to make sure that the IDPS module is always up to date. Since the UAVs are resource-constrained, online-learning (where the UAV updates its module at each step) would consume significant power. Hence, as an alternative, a centralized process was introduced to train the existing IDPS system on recent network traffic. An intelligent unit is placed in the central station, which contentiously uses the recent data relayed by the swarm of UAV to the station via cloud servers to train a global IDPS. By using such a method, the intelligent unit uses all the data exchanged by the swarm of UAVs to build a more generic and robust IDPS model. Unlike other IoT devices, UAVs should return to the charging station in cases of inactivity or battery depletion. Taking advantage of such criteria, an update to the UAV's IDPS module is scheduled every time it returns to the station, as described in Algorithm. 2.

Simply put, once an UAV returns to the docking station, the intelligent unit automatically makes sure to overwrite the old IDPS module implemented on the flying unit and replace it with a more recent module. The main advantage of the periodic offline-learning is to add, to the already trained IDPS module, new knowledge without forgetting what has already been learned. Additionally, compared to the online-learning, which is considerably infeasible in the case of UAVs, the periodic offline-learning process is cost-free, and still brings the same advantages to IDPS in terms of detection accuracy.

Algorithm 2: IDPS update process

```
Integrate_IDPS_Model(idps_p)
  // p^th version of the IDPS module
Employ UAVs fleet F. // Set of UAVs
foreach UAV u in F do
    b = current _batteryLevel(u)
    if b ≥ min_required_energy then
        Continue()
    else
        Return_to_dockingStation()
Procedure Return_to_dockingStation()
  Integrate_IDPS_Model(idps_{p+1})
  // p+1^th version of the IDPS module
b = b + charging_rate * charging_time
```

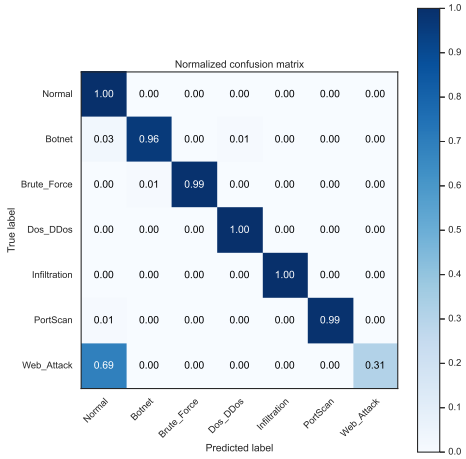

Fig. 2: Multi-class confusion matrix for the offline trained RL model (without periodic offline-learning).

V. Simulation Results

In this section, the behavior of the proposed IDPS module is studied. Also, a visualization of the impact of the periodic offline-learning on the efficiency of the module is presented. For this study, as previously mentioned, we use the CIC-IDS2017 dataset, which contains more than three million samples with fifteen unbalanced classes. To reduce this problem, new classes were formed, by merging few minority attack classes having similar characteristics and behavior as presented in [16]. Thus, reducing the total number of classes to just seven (1 normal + 6 attacks).

A. Accuracy

With the purpose of evaluating the performance of the proposed IDPS module and the used reward function discussed in section IV, the proposed RL-model was compared with

known algorithms and methodologies used for classification problems. The performances of the selected algorithms is presented in Table I. We can observe how the proposed IDPS module, especially with the customized reward function r_2, which encourages the model to detect minor classes with higher reward, produces good results, if not the best, along with the Random Forest model (taking into consideration the accuracy, F1-score and recall). The RL-based IDPS stands out in the Recall metric (in green). This metric is very crucial to the intrusion detection system, since it reflects the capability of differentiating between normal and malicious activities (minimum number of false negatives). Table I also shows a comparison between the efficiency of the two used reward functions. It is clear that the customized reward, which takes into consideration the unbalanced nature of our problem and the used data set, outperformed the simple [0, 1] function in all metrics.

TABLE I: IDPS module performance using different ML-techniques.

ML-classifier	Accuracy (%)	Precision	Recall	F-1
Logistic Regression	87.58	0.91	0.82	0.90
ANN	99.34	0.96	0.94	0.9
Random Forest	99.68	0.97	0.95	0.92
Naive Bayes	37.77	0.5	0.57	0.39
RL(r_1)	85.27	0.73	0.85	0.78
RL(r_2)	99.70	0.95	0.97	0.96

The notable performance of the RL(r_2) is supported by the confusion matrix given in Fig. 2, which highlights the intrusion detection efficiency of the proposed IDPS for almost all types of Cyber-attack presented in the used dataset. However, as illustrated in the Fig. 2, it is notable that the module has trouble in properly identifying one of the classes, namely 'Web-Attack'. This is explained by the fact that, during the training phase, the number of samples of this type of attack was insufficient for the module to extract enough information about the class (the number of samples was deliberately reduced to show the impact of the periodic offline learning). Consequently, to overcome such hurdle, the periodic-offline learning process presented in section IV was applied. Benefiting from the multiple updates introduced by the periodic offline-learning approach, after extracting sufficient information from the newly exchanged data, the IDPS module evolves and conquers new challenges.

In Fig. 3, we present the achieved detection percentage of the 'Web-Attack' versus the number of updates, using the periodic offline-training. It is noticeable that while increasing the number of introduced samples, the accuracy of the IDPS module increases as the module gains more knowledge about the introduced cyber-attacks. Figure. 3 also compares the performance of the IDPS, with and without the periodic-offline learning, which shows that the periodic-offline learning enabled the module to overcome this lacking, and became able to detect efficiently the 'Web-Attack' class.

B. Energy Consumption

A comparison between the performance of online learning and the proposed periodic offline-learning approaches was conducted in regards to the level of energy consumption and its impact on the lifetime of the UAVs. To do so, for this study, we assumed that the UAV power consumption consists primarily of two parts, namely, the power used by the UAV for hovering, and the power used by the UAV to establish a communication link and exchange data. With the purpose of using a realistic consumption model, we adopted the UAV power model presented in [17] to obtain the power consumption for hovering, referred to as P_{hov}. As stated, while receiving data, the UAV consumes power to maintain its altitude, namely the hovering power P_{hov}, in addition to the power used by the communication interfaces to exchange data, referred to as P_{com}, which is the needed power to communicate a message. Hence, the total power consumed P_{rx}, per second, during data reception is expressed as follows [3]:

$$P_{rx} = P_{hov} + P_{com} * M, \tag{3}$$

where M is the number of messages received in that time step (1 second).

We assume that the UAV exchanges its data with a data rate $R = 1mbps$, and by setting the average size of the messages exchanged by the UAV to $M = 500kb$, the UAV will receive a maximum of 2 messages per second. Adopting a scenario with a high matching degree to real-world applications, we randomized the process of data exchange. As such, the UAV receives a random number of $X \in [0, 2]$ messages per second.

In the proposed periodic offline-learning, the UAV should be concerned only by P_{rx}, since the learning process is made offline. However, for the case of online learning, a third component should be added to the power consumption equation. The power used to update the IDPS system should be considered. The power needed to update the model online is expressed as follows:

$$P_{lr} = P_{upd} * M, \tag{4}$$

where P_{upd} is the power used by the UAV to update its module after receiving M messages per second. Therefore, the total power consumed by the UAV, per second, is denoted by P_{tot} and expressed as follows:

$$P_{tot} = P_{lr} + P_{rx}. \tag{5}$$

Fig. 4 depicts the energy consumed by a UAV at the end of each cycle using online-learning in comparison to the energy used by the periodic offline-learning method while receiving the same number of messages. We also present the number of messages received by the UAV per cycle (presented on top of Fig. 4 in black). Results show that by using the online-learning approach, a UAV consumes more energy than the periodic offline-learning approach, which is explained by the additional power P_{lr} used by the UAV to update the IDPS module for the first approach. After six work cycles, as shown in the line

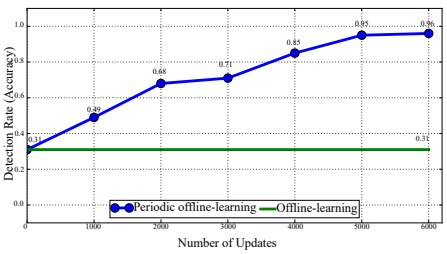

Fig. 3: Average detection percentage.

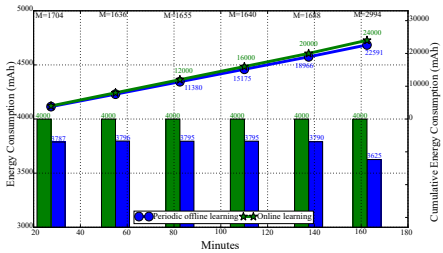

Fig. 4: UAV energy consumption.

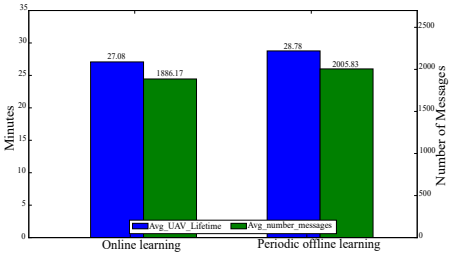

Fig. 5: Average effectiveness.

presentation which depicts the cumulative energy consumed by the UAV, the latter consumed a total of $2400mAh$ while using online learning, but only a total of $22591mAh$ while using our proposed approach to receive the same number of messages.

To further highlight the impact of the energy saved by periodic offline-learning on the lifetime and operation of the UAV, we carried out an effective comparison as depicted in Fig. 5. Using periodic offline-learning, the UAV lifetime, on average, is extended. The UAV operates approximately 29 minutes until the depletion of its battery using our proposed approach. However, it lasts for only 27 minutes per cycle while using the online-learning approach. A gain of $\approx$ 2 minutes might seem insignificant, but assuming that the UAV operates for around 48 cycles per day, there will be a gain of 96 minutes per day. To further manifest the importance of such gain, we compare the average number of exchanged messages per cycle. The UAV, using periodic offline-learning, manages to exchange $\approx$ 2000 messages per cycle. On the contrary, using online-learning, the number decreases to $\approx$ 1880. Ultimately, the periodic offline-learning is superior to the online approach in terms of power consumption, hence, minimum power consumption with better efficiency and operation time.

VI. Conclusion

In this paper, an efficient intrusion detection and prevention system for UAVs was developed while taking into consideration their limited resources. The flying units determine the security of the network using a reinforcement learning approach, by detecting suspicious activities and taking actions accordingly. A cost-effective periodic-offline learning approach is developed to ensure that the IDPS module is always up to date, capable of adapting to changes in attack patterns. The behavior of the proposed RL-based IDPS module for the different investigated scenarios validates the ability of the proposed IDPS to conduct the task of intrusion detection in the network efficiently. The simulation results show that periodic offline-learning efficiently enhances the accuracy of the IDPS module compared to the classic offline learning. Moreover, results show that the proposed module minimizes the energy consumed by the UAV compared to online-learning, and extends the UAV lifespan. In other words, better accuracy, with the minimal energy consumption.

References

[1] H. Dai, H. Zhang, C. Li, and B. Wang. Efficient deployment of multiple uavs for iot communication in dynamic environment. *China Communications*, 17(1):89–103, 2020.

[2] M. Aloqaily and et al. Design guidelines for blockchain-assisted 5g-uav networks. *IEEE Network*, 2020.

[3] O. Bouhamed, H. Ghazzai, H. Besbes, and Y. Massoud. A uav-assisted data collection for wireless sensor networks: Autonomous navigation and scheduling. *IEEE Access*, 8:110446–110460, 2020.

[4] O. Bouachir, M. Aloqaily, I. A. Ridhawi, O. Alfandi, and H. B. Salameh. Uav-assisted vehicular communication for densely crowded environments. In *IEEE/IFIP NOMS 2020*, 2020.

[5] M. Aloqaily, S. Otoum, I. Al Ridhawi, and Y. Jararweh. An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Networks*, 90:101842, 2019.

[6] H. Sedjelmaci, S. M. Senouci, and M. Messous. How to detect cyber-attacks in unmanned aerial vehicles network? In *IEEE GLOBECOM'16, Washington, DC, USA*, 2016.

[7] D. Muniraj and M. Farhood. A framework for detection of sensor attacks on small unmanned aircraft systems. In *ICUAS'17, Miami, USA*, 2017.

[8] H. Sedjelmaci, S. M. Senouci, and N. Ansari. A hierarchical detection and response system to enhance security against lethal cyber-attacks in uav networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9):1594–1606, 2018.

[9] M. P. Arthur. Detecting signal spoofing and jamming attacks in uav networks using a lightweight ids. In *International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2019.

[10] Samir Fenanir, Fouzi Semchedine, and Abderrahmane Baadache. A machine learning-based lightweight intrusion detection system for the internet of things. *Revue d intelligence artificielle*, 33:203–211, 10 2019.

[11] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo. Toward a lightweight intrusion detection system for the internet of things. *IEEE Access*, 2019.

[12] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri. A lightweight anomaly detection technique for low-resource iot devices: A game-theoretic methodology. In *IEEE International Conference on Communications (ICC'16), Kuala Lumpur, Malaysia*, pages 1–6, 2016.

[13] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud. Scalable and secure architecture for distributed iot systems. In *IEEE Technology Engineering Management Conference (TEMSCON'20), Novi, MI, USA*, 2020.

[14] O. Bouhamed, H. Ghazzai, H. Besbes, and Y. Massoud. A generic spatiotemporal scheduling for autonomous uavs: A reinforcement learning-based approach. *IEEE Open Journal of Vehicular Technology*, 2020.

[15] Y. Zeng, R. Zhang, and T. J. Lim. Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Communications Magazine*, 54, 2016.

[16] Ranjit Panigrahi and Samarjeet Borah. A detailed analysis of cicids2017 dataset for designing intrusion detection systems. 7:479–482, 01 2018.

[17] Y. Zeng and et al. Energy minimization for wireless communication with rotary-wing UAV. *IEEE Trans. Wireless Commun.*, 2019.