

Analyse or Transmit

Utilising Correlation at the Edge with Deep Reinforcement Learning

Hribar, Jernej ; Shinkuma, Ryoichi ; Iosifidis, George; Dusparic, Ivana

DOI

[10.1109/GLOBECOM46510.2021.9685166](https://doi.org/10.1109/GLOBECOM46510.2021.9685166)

Publication date

2021

Document Version

Final published version

Published in

2021 IEEE Global Communications Conference (GLOBECOM)

Citation (APA)

Hribar, J., Shinkuma, R., Iosifidis, G., & Dusparic, I. (2021). Analyse or Transmit: Utilising Correlation at the Edge with Deep Reinforcement Learning. In *2021 IEEE Global Communications Conference (GLOBECOM): Proceedings* (pp. 1-6). Article 9685166 IEEE. <https://doi.org/10.1109/GLOBECOM46510.2021.9685166>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' – Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Analyse or Transmit: Utilising Correlation at the Edge with Deep Reinforcement Learning

Jernej Hribar*, Ryoichi Shinkuma[†], George Iosifidis[‡], and Ivana Dusparic*

*CONNECT-Trinity College Dublin, Ireland, Email: {jhribar, duspari}@tcd.ie

[†]Faculty of Engineering, Shibaura Institute of Technology, Japan, Email: shinkuma@shibaura-it.ac.jp

[‡]Delft University of Technology, Netherlands, Email: g.iosifidis@tudelft.nl

Abstract—Millions of sensors, cameras, meters, and other edge devices are deployed in networks to collect and analyse data. In many cases, such devices are powered only by Energy Harvesting (EH) and have limited energy available to analyse acquired data. When edge infrastructure is available, a device has a choice: to perform analysis locally or offload the task to other resource-rich devices such as cloudlet servers. However, such a choice carries a price in terms of consumed energy and accuracy. On the one hand, transmitting raw data can result in a higher energy cost in comparison to the required energy to process data locally. On the other hand, performing data analytics on servers can improve the task's accuracy. Additionally, due to the correlation between information sent by multiple devices, accuracy might not be affected if some edge devices decide to neither process nor send data and preserve energy instead. For such a scenario, we propose a Deep Reinforcement Learning (DRL) based solution capable of learning and adapting the policy to the time-varying energy arrival due to EH patterns. We leverage two datasets, one to model energy an EH device can collect and the other to model the correlation between cameras. Furthermore, we compare the proposed solution performance to three baseline policies. Our results show that we can increase accuracy by 15% in comparison to conventional approaches while preventing outages.

Index Terms—Deep Reinforcement Learning, Green Communications, Energy-harvesting, Edge Computing, Data-analytics

I. INTRODUCTION

In the last few years, we have witnessed unprecedented progress in Machine Learning (ML) methods that paved the way for many new analytic services where embedded devices collect and process data to extract information of interest [1]. Coupled with the rise of available processing power at the network edge through cloudlet servers [2], applications that require data analysis such as cameras counting objects or sensors detecting anomalies have become the new norm. In such a scenario, devices are confronted with a decision: to analyse collected data locally (and transmit only the extracted information of interest) or transmit obtained data to a fusion centre for processing. Such a decision carries a cost in terms of device's energy consumption and applications' performance, e.g., the accuracy of detected anomaly.

Providing energy necessary to power edge devices is becoming increasingly more challenging [3]. While connecting devices to the power grid is the most reliable way of providing energy, it also results in high deployment costs. On the one hand, Energy Harvesting (EH) is proving a viable alternative as it enables the device to collect energy by tapping into the various ambient sources such as wind, solar radiation,

vibrations, etc. [4]. On the other hand, a device with an EH has to be aware of its energy usage while ensuring that the application performs as required. Additionally, the collected energy by an EH varies over time. For example, during the day, a device with a solar panel can gather plenty of energy, while during the night, collected energy will be negligible.

In large sensor networks, the information obtained from individual devices is often correlated. For example, devices might be monitoring overlapping areas and multiple devices can detect the same anomaly or the observing events are correlated. Therefore, redundancy creates an opportunity to reduce the operational costs of deployments. In other words, an EH-powered device can take advantage of the correlation in order to improve its energy efficiency. For example, a device can choose not to take any actions and preserve energy instead, as another device will obtain the required information. However, to enable such an approach, the system has to learn how to take advantage of correlated information.

Existing work mostly focuses on a single aspect of this problem. For example, [5]–[8] consider offloading computational task from an EH-powered device to a server, but without considering the accuracy of the task. Work in [9]–[11], on the other hand, considers data-analytics scenarios, but does not consider EH-powered devices. In addition, neither of the above approaches consider taking advantage of correlation.

To address this gap, in this paper we investigate the impact of correlation in the outcome of the data-analytic task on the energy efficiency of EH-powered devices at the network edge. To examine the characteristics of such a system, we focus on a case study of a set of cameras powered by EH counting traffic in an intersection. Cameras can choose to perform object detection or transmit a raw image to be processed on a nearby cloudlet server. Additionally, due to correlation, one of the cameras can decide not to transmit or perform object detection because the system will be able to count vehicles accurately using an image from the other camera. In such a scenario, object detection can be performed by only a subset of all available cameras while it is still possible to detect all objects. The system's goal is to count observed objects accurately, which we measure through a standard ML metric recall while being limited by the time-varying energy of cameras with EH. We identified Deep Reinforcement Learning (DRL) as the most suitable approach to design an autonomous solution capable of adapting to varying energy of a device

with EH while taking advantage of correlated information. The benefits of learning approach in such environments was demonstrated by many other DRL-based solutions [12]–[15]. The contributions in this paper can be summarised as follows:

- We propose and formulate the problem of joint energy and accuracy optimization in EH edge networks with correlated sources of information.
- We propose a novel DRL-based algorithm that guides the decision for EH-powered sensors based on available energy, task accuracy, and correlation.
- We enrich the DRL solution with a Gaussian Processes (GP)-based module for predicting on the fly the devices' energy availability.
- We evaluate the performance of the two variants of the DRL-based solutions (with and without GP) using two established datasets [16], [17] to simulate the environment as accurately as possible and compare our solution to three baseline policies.

II. RELATED WORK

Our work is related to studies that proposed task offloading schemes for edge devices powered by an EH [5]–[8]. For example, the work in [5] leverages DRL to decide which edge device will offload the task so as to minimize their energy consumption and task latency. The authors in [6] propose an architecture and a threshold policy to achieve energy-aware edge task offloading from EH sensors. Similarly in [7], the authors rely on online Lyapunov based task offloading algorithm to investigate the trade-off between energy consumption and execution delay. And [8] considered online learning to decide on how much power should a Mobile Edge Computing (MEC), powered by an EH, allocate to each task. However, the above solutions do not optimise the accuracy of the data-analytics task, nor aim to design a system that will ensure that EH powered devices will avoid depleting all available energy as is the objective of our work.

Only a handful of papers examine the optimisation of task offloading schemes to improve the accuracy of data-analytics [9]–[11]. In [9] the authors use facial recognition as an example to validate the energy effectiveness of their proposed task offloading scheme. However, in their system EH is not considered. Similarly, a trade-off between object detection accuracy and latency was explored in [10] for battery-powered mobile devices. The focus of the latter was to demonstrate that task offloading can improve frame rate and accuracy. An online learning approach was proposed in [11], which improves the accuracy of the data-analytics task while reducing the energy consumption of an edge device. In contrast, we explore a more advanced system in which devices are powered only by an EH. In our system, when deciding the system must also consider the energy cameras will collect in future and correlation between cameras to achieve optimal performance.

Recently, Reinforcement Learning (RL) emerged as a very effective tool to resolve a plethora of problems related to the management of EH devices due to its ability to adapt to

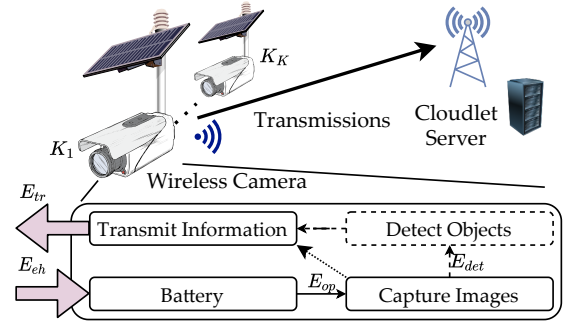


Fig. 1: Illustration of the system model and the main wireless camera components.

a dynamic environment [13]–[15]. For example, in [13] the authors proposed a RL-based power management capable of maximising the quality of service while considering available energy and energy cost of transmission. In [14] the authors employed DRL to resolve an access problem for EH devices. A balance between transmission power and modulation level to increase throughput was considered in [15]. In the papers reviewed above, DRL was selected because of its adaptability to the time-varying nature of energy arrival on EH-powered devices, leading us to consider it as a suitable approach for the problem our paper is addressing as well.

III. SYSTEM MODEL AND PROBLEM STATEMENT

We consider a network consisting of K wireless cameras powered by an EH and a single cloudlet server as illustrated in Fig. 1. The cameras are embedded devices capable of processing frames, i.e., images, using dedicated hardware, and transmitting gained information to the cloudlet server for collection. Alternatively, a camera can transmit captured frames and allow the cloudlet server to process them. We assume that time is slotted $t \in \mathcal{T} = \{0, 1, \dots, T-1\}$ and at each time slot each cameras captures one image. The system updates its decision every τ time-steps as due to relatively high cameras frame rate¹ λ deciding every time-step would be impractical.

A. Cameras' Energy Parameters

In each time-step, the k -th camera receives harvested energy $E_{eh}^{(k)}(t)$ proportional to the current $I_{eh}^{(k)}(t)$ and the voltage $U_{eh}^{(k)}(t)$ from the Energy Harvesting (EH) unit. To store the captured energy, each cameras is quipped with a battery that has a maximal capacity E_{max} . Meaning, camera's available energy $E_{av}^{(k)}(t)$ is limited to an interval $E_{av}^{(k)} \in [0, E_{max}]$. Note that if camera's energy is zero, i.e., $E_{av}^{(k)}(t) \leq 0$, camera turns-off and no images nor detected objects are available to the system. We denote the downtime of k -th camera as $T_{down}^{(k)}$.

Operating the camera consumes $E_{op}(t)$ of energy at each time step. This energy is required to support the camera's essential operation, such as capturing images. We assume that the system has no control over $E_{op}(t)$ energy consumption regardless of how the camera operates. Therefore, even if the

¹Typically traffic video cameras capture 10 frames per second [18]

system decides that no images are required from the camera, the camera will still consume $E_{op}(t)$. The $E_{op}(t)$ depends mainly on the hardware used. In the validation section, we model $E_{op}(t)$ according to the consumption of a Raspberry Pi with a camera.

Whenever the camera decides to transmit the image, the energy consumption is relative to the raw image size in bits we denote as $f_{raw}^{(k)}(t)$ multiplied by the energy cost of transmitting a bit of information $E_{tr}(t)$. To detect the object locally, we assume that the camera consumes $E_{det}(t)$ of energy. Such energy is required to process the image. Furthermore, the camera will also consume a small amount of energy to transmit obtained information, e.g., the number of detected objects. We denote the size of such data packet with $f_{proc}^{(k)}(t)$. However, the latter data packet is much smaller in size than when camera decides to transmit raw image, i.e., $f_{raw}^{(k)}(t) > f_{proc}^{(k)}(t)$.

B. Object Detection, Correlation, and Recall

We define $\mathcal{C}(t)$ as the set of all objects present at a particular time-step t in the scenery of interest, e.g., the intersection. A camera can not always detect all objects due to practical limitations, thus we define a set of all possible objects a camera can detect as $\mathcal{C}_k(t)$ which is a subset of $\mathcal{C}(t)$, i.e., $\mathcal{C}_k(t) \subseteq \mathcal{C}(t)$. We assume that a device can process captured images using dedicated hardware capable of efficiently processing image. Edge TPU, NVIDIA Xavier, and NovuTenso are examples of such dedicated hardware units capable of ensuring that the energy cost of processing on edge device is lower than the energy cost of transmitting captured images [19]. To detect objects of interest, the system can choose to employ object detector J_{yolo} located on the cloudlet or J_{tiny} on the camera. We use You Only Look Once (YOLO) version 3 [20] for object detection on the cloudlet and tiny-yolo version 4 to perform object detection on the camera. Using one of the available detectors, the system can then obtain a set of objects detected by the individual cameras as follows:

$$J_{yolo} : \mathcal{C}_k(t) \rightarrow \mathcal{K}_k(t) \text{ or } J_{tiny} : \mathcal{C}_k(t) \rightarrow \mathcal{K}_k(t), \quad (1)$$

where $\mathcal{K}_k(t)$ is the set of object that are detected in the image of camera k at time step t . Note, that if camera is in the stand-by mode or has no available energy, the resulting set is empty. Finally, we can define the set of all the objects the system has detected as $\mathcal{K}(t) = \mathcal{K}_1(t) \cup \mathcal{K}_2(t) \dots \cup \mathcal{K}_K(t)$. We also define the set of correlated objects as $\mathcal{C}_\rho(t) = \mathcal{C}_1(t) \cap \dots \cap \mathcal{C}_K(t)$, i.e., the set of objects that all cameras can detect.

To measure the performance of the system we adopt recall, a standard metric in ML applications used to measure how well can a system detect relevant objects. In our case, the relevant objects are in $\mathcal{C}(t)$. Therefore, we define recall as:

$$\phi(t) = \frac{|\mathcal{K}|}{|\mathcal{C}(t)|}. \quad (2)$$

Recall is limited to an interval $\phi(t) \in [0, 1]$. The higher the recall value the more objects the system detected.

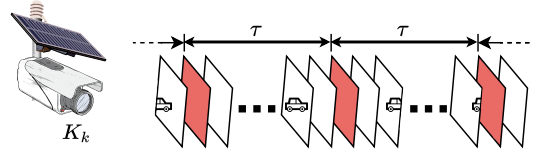


Fig. 2: At each τ decision interval, all cameras will transmit the raw image to the cloudlet server.

C. Problem Formulation

The main objective of the system is to find a policy π , that decides where or if should camera's image be processed, that will maximise recall ϕ . We formulate the problem as:

$$\begin{aligned} \max_{\pi} \quad & \bar{\phi}(t) \\ \text{s.t.} \quad & E_{av}^{(k)}(t) \geq 0, \forall k = 1, \dots, K \\ & T_{down}^{(k)} \leq 0, \forall k = 1, \dots, K \\ & t \in \mathcal{T} = \{0, 1, \dots, T-1\} \end{aligned} \quad (3)$$

The main constraint that policy faces is limited available energy which varies over time. Simultaneously, the system minimise cameras' outages, i.e., $T_{down} = 0$.

IV. DRL-BASED SOLUTION

Finding a policy capable of maximising accuracy and preserve cameras' energy in a time-varying environment is a non-trivial task. Therefore, we base our solution on DRL, which recently emerged as an ideal tool for optimising the system's performance in such an environment [12] due to its adaptability. Furthermore, the system also has to overcome a practical challenge. In deployments, $\mathcal{C}(k)$ is unknown to the system. To tackle the issue, we propose the use of the guard interval, i.e., at every decision epoch τ all cameras transmit a raw image as illustrated in Fig. 2. The system then utilises J_{yolo} to determine the $\mathcal{K}(\tau)$, which for the proposed solution represents the real state of the system on which all decisions are based. In our preliminary studies in which we analysed the video dataset [16] we discovered that using the J_{yolo} on images from all available cameras at time t results in $\bar{\phi}(t) = 0.99$. We then use the information extracted from guard images to determine the states, actions, and rewards, i.e., a tuple in $\langle \mathcal{S}, \mathcal{A}, R \rangle$.

A. States, Actions, and Reward

The state $\mathbf{s}(t) \in \mathcal{S} \subseteq R^{K+2}$ comprises the residual energy in each device, the guard recall ($\phi_g(t)$) and the number of the detected objects, i.e., $|\mathcal{K}(\tau)|$. We determine the $\phi_g(t)$ by comparing the number of objects we counted one time-step before the guard interval as follows:

$$\phi_g(\tau) = \frac{|\mathcal{K}(\tau-1)|}{|\mathcal{K}(\tau)|}. \quad (4)$$

Such a state space, even with a low granularity of discretising the inputs, rises extremely quickly to a thousands of states. For example, even using only a hundred states for the energy level, and relatively low granularity of 0.025 for the recall state, for

$K = 4$, the number of possible states is in the millions. This is a rationale for designing a solution based on deep rather than tabular RL.

Each camera has three different modes of operation: transmit raw image to the cloudlet for processing, use local object detector, or enter stand-by mode. Each action a in the set of available actions \mathcal{A} represent a vector with dimension K , consisting of one operation mode per sensor.

The reward consists of two parts. The first part is based on the observed recall value in a time-step before the decision time-step. To distinguish between acceptable rates and unsatisfactory ones, this reward is expressed as $r_\phi = 10(\phi_g(\tau) - 0.5)$, meaning that any recall less than 0.5 results in a negative reward, while only those higher than 0.5 result in a positive one. To further accelerate learning, the resulting reward is multiplied by 10. The second part of the reward depends on the cameras' energy; if any of the cameras have less than 15% the agent receives a high negative reward.

B. Estimating Energy and Implementation

To estimate the energy the camera collects through EH we use Gaussian Processes (GP) [21]. Namely we selected a standard periodic kernel in combination with a white noise kernel. We fit GP model using measurements of $I_{eh}^{(k)}(t)$ the system can easily access. Additionally, fitting the GP model for estimation can be carried out only once per day thus adding minimal overhead in terms of required processing power. However, the dimension of the input state vector $s(t)$ increases to $2K + 2$, i.e., $\mathcal{S} \subseteq R^{2K+2}$. The new state space has to also encompass the information regarding the estimated energy. In the next section, we demonstrate that using GP to estimate energy can greatly aid the system in reducing the T_{down} while also positively impacting recall.

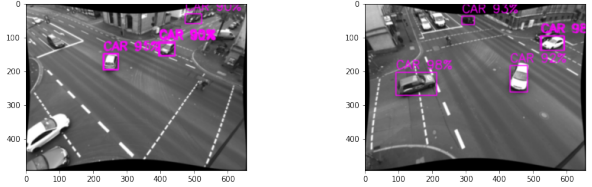
TABLE I: DQN Hyperparameters

Hyperparameter	Value	Hyperparameter	Value
Batch size	128	Memory size	$2 * 10^5$
Optimizer	Adam	Loss Function	MSE
Target ANN soft update τ	10^{-3}		

We implemented the learning agent in the cloudlet server where enough computational power is available to support processing required to support Deep Q-Network (DQN) and GP. Our DQN implementation uses an Artificial Neural Network (ANN) that consist of four hidden layers. The first and the last hidden layer have four neurons, while the middle two layers have eight neurons. We use the ReLU activation function for all layers except for the output layer, activated with a linear function. To prevent over-fitting, we apply a ten percent dropout between the layers. We list the rest of the hyperparameters in Table I.

V. VALIDATION AND RESULTS

In this section, we validate the performance of the proposed solution using real observations from two datasets. We test two variations of our DRL-based approach, one with estimated arrival energy we obtain using GP and one without such



(a) Camera one.

(b) Camera two.

Fig. 3: Example of images obtained from Ko-PER dataset at the same time-step from both cameras and objects detected by J_{yolo} .

information. For comparison, we designed three heuristic approaches: greedy, threshold, and alternating. We evaluate the performance in two scenarios: static and dynamic varying, cost of transmitting a bit. We simulate the performance over fifteen days, but the reported results are based on the average of the last twelve days. Our experiments show that the DRL-based solution will start outperforming a random approach within one day and then requires another two to three days to converge to the best policy the agent can learn. Additionally, to minimise the impact of randomness on results, we report the average of five iterations for each tested approach.

A. Use of Real-data and Heuristic Approaches

The Ko-PER intersection dataset [16] comprises of monochrome camera images and raw laser scanner measurements for an intersection. The data sequence is six and half minutes long or 9670 frames obtained from two different viewpoints, i.e., cameras. Consequently, in this evaluation, we had to limit the number of cameras to two, i.e., $K = 2$. Furthermore, we extrapolate the data by re-sampling. The Ko-PER images are obtained with 25 frames per second. However, in our simulation, we set the λ to ten as is the standard for traffic camera [18]. To avoid correlation between the samples, every minute we randomly switch the starting point in the sequence from which our simulation receives frames. Fig. 3 shows an example of how our system counts vehicles. To obtain $\mathcal{C}(t)$, we manually counted vehicles inside the intersection. Note that only the white vehicle in the bottom left, as seen from camera one, i.e., Fig. 3(a), has entered the intersection. The same applies to vehicles on top of the image from camera two, i.e., Fig. 3(b), those vehicles have already left or have not entered the intersection yet. Meaning that $\mathcal{C}(t) = 3$. The resulting $\mathcal{K}_1(t)$ and $\mathcal{K}_2(t)$ would be two and three, respectively.

We obtain the $I_{eh}^{(k)}(t)$ values from the energy-harvesting testbed installed on UCLM, Ciudad Real (Spain) [17]. We use measurements obtained between the 16th and 30th of August 2018 as input to our simulation, which enables us to test the performance of the proposed scheme over 15 days. The nominal power of used solar panels is $2W$, and in practice, it is expected that multiple panels are used to power a wireless camera. We assume that each of the cameras is equipped with four panels with 80% efficiency, i.e., $\eta_{eh} = 0.8$. We list the rest of the simulation parameters in Table II which we modelled according to existing measurements. For example, E_{det} we determined according to measurements

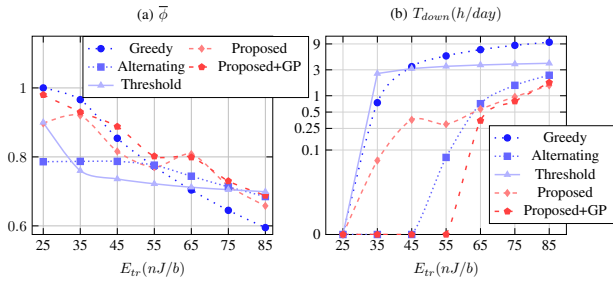


Fig. 4: Recall and downtime depending on the energy cost of transmission for different approaches.

in [19]. Furthermore, we also use real measurement to fit the GP model to estimate collected energy. After performing empirical studies considering the intervals for model fitting and estimated energy, we determined that using the past seven days of data and estimating the amount of energy for the next six hours is most effective. The resulting approach can very accurately estimate the amount of energy a camera will collect in the future².

TABLE II: Simulation Parameters

Parameter	Value	Parameter	Value	Parameter	Value
K	2	λ	10	τ	1s
$U_{eh}^{(k)}(t)$	6.1V	E_{max}	185kJ	$E_{op}^{(k)}$	139mJ
η_{eh}	0.8	E_{det}	57.48mJ	τ	1s

We compare the performance of our proposed solution to three heuristic approaches:

- 1) **Greedy:** When cameras adopt greedy policy they will always transmit the image, provided sufficient amount of energy is available.
- 2) **Threshold:** The camera will transmit the image if it has more than half of the overall energy available, i.e., $E_{av}(t)^k E_{max}$. Otherwise, the camera will perform object detection locally and then transmit results.
- 3) **Alternating:** Cameras iterate over \mathcal{A} set of actions. At each decision epoch τ the operating mode of cameras changes.

B. Static Cost of Transmission

Fig. 4(a) shows the average recall value the system achieves depending on the energy cost of transmission. Both of our proposed solutions have a very good performance in terms of achieved recall. When the cost of transmission increases, the performance of all approaches performance decreases, as expected. The recall is lower when the cost of transmission is high because the proposed solution must more often select an action that consumes less energy, resulting in a lower chance of correctly detecting a vehicle. Note that for baseline approaches such as greedy or alternating, the recall value lowers due to increased T_{down} , as shown in Fig. 4(b). For example, the greedy policy will result in the best possible average recall when the cost of transmitting is low. The opposite happens to

²A python notebook with detailed analysis can be found here: github.com/hribarjernej89/estimating-eh-energy-with-GP.

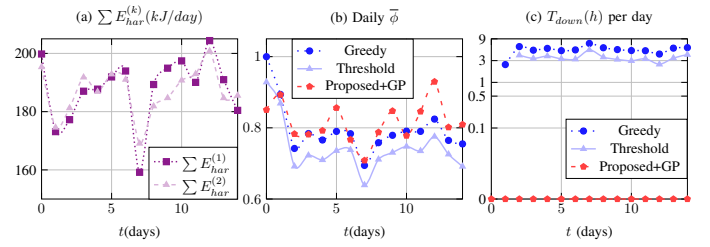


Fig. 5: The impact of daily collected energy on the performance of the system over a number of days.

the greedy policy when the energy cost is high, and cameras experience downtime of nine hours per day.

By comparing the performance of the proposed solutions to the alternating policy, we show that by selecting policy intelligently, it is possible to increase recall while simultaneously lower T_{down} as shown in Fig. 4(b). In its essence, the alternating policy represents results a system would obtain if actions would be selected randomly. On the other hand, the threshold policy seems reasonable. Still, it does not perform well when the cost of transmission is low and also leads to a relatively high daily downtime when the energy cost of transmission is high. The difference between the two proposed solutions is that one of them receives additional information in a form of estimated energy. Consequently, while the estimated energy has minimal impact on the achieved recall value, the T_{down} is significantly reduced in comparison when we use GP to estimate collected energy.

C. Dynamically Varying the Cost of Transmission

Next, we observe recall and daily downtime as the energy cost of transmission randomly varies over time. At every decision epoch the energy cost E_{tr} is randomly sampled from an interval between $25nJ$ and $85nJ$, i.e., $E_{tr} \in [25, 85]nJ$. In Fig. 5 we show how the recall and T_{down} change over several days for three selected policies: greedy, proposed with added estimated energy using GP, and threshold. In Fig. 5(a) shows how much energy each camera collects in a day. On average, camera one collects more energy. However, the difference is minimal. In Fig. 5(b), we show recall value. Our solution is better at compensating days when cameras obtain less energy, e.g., both cameras collect significantly less energy on day seven than on other days. In Fig. 5(c) we plot downtime. While the proposed solution results in zero downtime, the greedy approach and threshold directly correlate to collected energy, a trend that is especially noticeable on day seven.

Fig. 6 presents the achieved daily values of average recall and downtime for each policy when the cost of transmissions randomly varies. The two proposed solutions acquire the best recall values. However, daily variations are lower when we use DRL-based approach with estimated energy values. Interestingly, the threshold policy results in the worst recall values. We plot daily downtime in Fig. 6(b). The greedy approach results in the highest downtime per day. Interestingly, the greedy achieves recall similar to the alternating policy. The latter has almost a negligible downtime of only three minutes per day. Similarly, we could state for our proposed

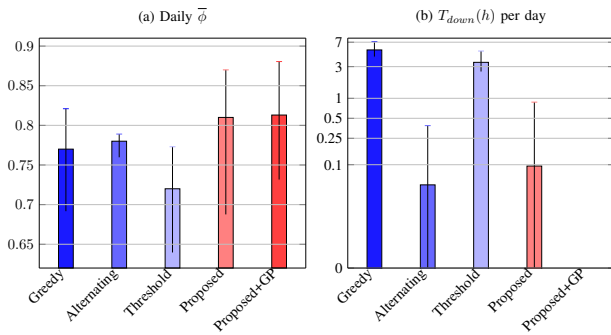


Fig. 6: Achieved daily average recall and downtime for all approaches when the energy cost of transmission randomly varies over time.

approach as it results in a downtime of around 6 minutes per day. Nevertheless, when we add additional information regarding the incoming energy, the proposed solution will result in no downtime whatsoever. In practice, the system always needs some energy to act in case of an emergency. Therefore, the proposed solution in combination with GP is the most beneficial as it will ensure that system will always have some energy left to operate.

VI. CONCLUSION

In this paper, we studied how can sensor network take advantage of correlation to improve the energy efficiency of EH-powered devices without impacting performance. We showed the benefits of utilising correlation in a scenario of multiple cameras powered only by EH, observing the same intersection with a goal to count the number of vehicles accurately. We proposed a DRL-based approach to select cameras' actions to improve system performance we measure through recall while preventing outages. Our results show that the proposed approach can be up to 15% more accurate with no outages. With millions of EH powered devices in future networks, the use of correlated information will become increasingly more important to balance the energy performance of such devices while maximising their performance.

In our future work, we will examine the scalability of our solution. In theory, the more cameras the system has, the more advantageous should it be to leverage correlation. However, as the number of cameras increases, so does the number of available actions; thus, a solution capable of dealing with a high number of actions is required. We will also investigate the delay. The camera needs more time to process the image than a cloudlet server. In contrast, it takes more time to transmit raw image than the result of the data analytics task. By considering the delay, we will add another dimension to our work. Therefore, we expect to observe interesting results when considering three tradeoffs: energy, accuracy, and delay.

ACKNOWLEDGEMENTS

This work was funded in part by the European Regional Development Fund through the SFI Research Centres Programme under Grant No. 13/RC/2077_P2 SFI CONNECT, the SFI-NSFC Partnership Programme Grant No. 17/NSFC/5224,

and European Union's H2020 research and innovation programme under grant agreement DAEMON 101017109. This work was also supported in part by JST PRESTO Grant No. JPMJPR1854, JSPS KAKENHI Grant No. JP21H03427, and JSPS International Research Fellow Grant No. PE20723.

REFERENCES

- [1] E. Siow, T. Tiropanis, and W. Hall, "Analytics for the Internet of Things: A Survey," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, 2018.
- [2] J. Chen and X. Ran, "Deep Learning With Edge Computing: A Review," *Proced. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.
- [3] A. P. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, "A Survey of Green Networking Research," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 1, pp. 3–20, 1st Quart., 2012.
- [4] S. Bi, C. K. Ho, and R. Zhang, "Wireless Powered Communication: Opportunities and Challenges," *IEEE Commun. Magazine*, vol. 53, no. 4, pp. 117–125, Apr. 2015.
- [5] M. Min, L. Xiao, Y. Chen, P. Cheng *et al.*, "Learning-based Computation Offloading for IoT Devices with Energy Harvesting," *Trans. Veh. Tech.*, vol. 68, no. 2, pp. 1930–1941, 2019.
- [6] V. Balasubramanian, N. Kouvelas, K. Chandra, R. V. Prasad *et al.*, "A Unified Architecture for Integrating Energy Harvesting IoT Devices with the Mobile Edge Cloud," in *Proc. IEEE WF-IoT*. Singapore, 2018, pp. 13–18.
- [7] W. Liu, X. Zhou, S. Durrani, H. Mehrpouyan, and S. D. Blostein, "Energy Harvesting Wireless Sensor Networks: Delay Analysis Considering Energy Costs of Sensing and Transmission," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4635–4650, 2016.
- [8] J. Xu, L. Chen, and S. Ren, "Online Learning for Offloading and Autoscaling in Energy Harvesting Mobile Edge Computing," *IEEE Trans. Cog. Comms. Net.*, vol. 3, no. 3, pp. 361–373, Sep. 2017.
- [9] X. Lyu, H. Tian, L. Jiang, A. Vinel *et al.*, "Selective Offloading in Mobile Edge Computing for the Green Internet of Things," *IEEE Network*, vol. 32, no. 1, pp. 54–60, 2018.
- [10] X. Ran, H. Chen, Z. Liu, and J. Chen, "Delivering Deep Learning to Mobile Devices Via Offloading," in *Proc. ACM SIGCOMM Workshop*. Los Angeles, CA, USA, 2017, pp. 42–47.
- [11] A. Galanopoulos, G. Iosifidis, and T. Salonidis, "Optimizing Data Analytics in Energy Constrained IoT Networks," in *Proc. IEEE WiOpt*. Shanghai, China, May 2018, pp. 1–8.
- [12] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato *et al.*, "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019.
- [13] F. A. Aoudia, M. Gautier, and O. Berder, "RLMan: An Energy Manager Based on Reinforcement Learning for Energy Harvesting Wireless Sensor Networks," *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 2, pp. 408–417, Jun. 2018.
- [14] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement Learning-based Multiaccess Control and Battery Prediction With Energy Harvesting in IoT Systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2009–2020, 2018.
- [15] M. Li, X. Zhao, H. Liang, and F. Hu, "Deep Reinforcement Learning Optimal Transmission Policy for Communication Systems With Energy Harvesting and Adaptive MQAM," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5782–5793, 2019.
- [16] E. Strigel, D. Meissner, F. Seeliger, B. Wilking, and K. Dietmayer, "The Ko-PER Intersection Laserscanner and Video Dataset," in *Proc. IEEE ITSC*. Qingdao, China, Oct. 2014, pp. 1900–1901.
- [17] M. Kuzman, X. del Toro Garcia, S. Escolar, A. Caruso *et al.*, "A Testbed and an Experimental Public Dataset for Energy-Harvested IoT Solutions," in *Proc. INDIN*, vol. 1, Jul. 2019, pp. 869–876.
- [18] M. Naphade, Z. Tang, M.-C. Chang, D. C. Anastasiu *et al.*, "The 2019 AI City Challenge," in *Proc. CVPR Workshops*, 2019, pp. 452–460.
- [19] Y. Hui, J. Lien, and X. Lu, "Early Experience in Benchmarking Edge AI Processors with Object Detection Workloads," in *Proc. Bench 2019*. Springer, 2019, pp. 32–48.
- [20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE CVPR*. Las Vegas, NV, USA, June 2016, pp. 779–778.
- [21] S. Roberts, M. Osborne, M. Ebdon, S. Reece *et al.*, "Gaussian Processes for time-series modelling," *Phil. Trans.*, vol. 371, no. 1984, 2013.