Selective Data Offloading in Edge Computing for Two-Tier Classification With Local Domain Partitions

Forough Shirin Abkenar⋆, Leonardo Badia†, and Marco Levorato⋆
⋆ Donald Bren School of Information and Computer Sciences, University of California at Irvine, United States
† Dept. of Information Engineering (DEI), University of Padova, Italy

***Abstract*—We consider a two-tier approach for the classification of user-generated data, where low-complexity decision algorithms are available on mobile devices, and a better assessment can be performed on a shared edge server to which the samples can be offloaded. While an overall accurate classification can be achieved by either massive offloading to the edge server alone or performing a computationally intense domain partitioning for local evaluation, both these solutions taken individually are excessively demanding. Importantly, the former strategy achieves higher accuracy, yet is very bandwidth-consuming, while the latter results in lower accuracy while reducing bandwidth usage. To cope with these challenges, we take a quantitative stance to investigate the benefit of combining these two strategies, i.e., performing most of the evaluations with a local decision over constrained domains, while at the same time offloading to the edge server a small fraction of the samples for which the classification is expected to be less accurate. If properly harmonized, such an approach is shown to lead to a sharp increase in classification accuracy, with overall limited resource usage, which makes it suitable for practical implementations.**

***Index Terms*—Edge computing, Supervised learning, Domain adaptation.**

I. Introduction

The real-time processing of information-rich samples is an important challenge, that can be solved only by bridging the pervasive sensing capabilities of the Internet of things (IoT) with data-efficient analysis and machine learning (ML) [1]. Potential applications include a plethora of scenarios such as manufacturing, automotive, and smart healthcare. The hurdle between massive data acquisition and proper exploitation for decision-making (DM) lies in that the involved artificial intelligence techniques, such as deep neural networks, are often complex and resource-hungry, which is at odds with the limitations of mobile devices (MDs), in terms of energy, computation, and memory [2], [3].

Many industrial and research applications exploit two common solutions to tackle such an issue, namely: (i) simplification or compression of the ML models to account for the constraints of the MDs [4], [5]; and (ii) offloading the computing tasks to more powerful devices located at the network edge [6], [7], [8]. We claim that neither of these two approaches is ideal for exclusive adoption in some settings, as the former is prone to performance degradation, while the latter may use excessive channel resources [9].

For this reason, we investigate the benefits of a hybrid solution within an online semantic domain-restricted classification. The idea is to perform the classification of a wide set of samples by making use of classifiers at two tiers, either locally or in the ES. Thus, we define the local-scaled domains, named the local classification domains (LCDs). An LCD is a confined area of the feature space that incorporates a subset of samples for local classification [10]. In the binary classification case considered herein, we use simpler linear classifiers with LCD-specific parameters to approximate the decision boundary. The adaptation improves accuracy with respect to a global approximation, but may not entirely eliminate the need for offloading some of the samples for a more accurate classification at the ES.

Both solutions can individually lead to high accuracy if pushed to the extreme, i.e., exploiting many LCDs or offloading the entire set of samples to the ES, but these extreme points are often impractical due to their high resource demands. Yet, we argue that the highest incremental benefit of each individual technique is achieved with a relatively limited usage, which prompts us to investigate whether it is actually convenient to identify a resource-efficient harmonization of the two approaches. As a result, the present paper investigates the development of a hybrid approach where both strategies are used in a balanced way. We evaluate the practicality of combining offloading to an ES of a small part of the samples while exploiting the limitations of LCDs with low-complexity classifiers [11]. Moreover, for some specific applications, it may be convenient to only focus on potential false negatives [12], which is explored as a way to improve performance.

To summarize, this paper makes the following contributions. First, we develop a combined technique for domain-constrained classification with proper exploitation of the edge computing architecture. Moreover, we investigate from a quantitative standpoint the performance of using local classifiers with properly trained domains versus offloading to the ESs, and we evaluate the total benefit that can be achieved by combining the two techniques. Additionally, we discuss the role of a proper selection of the samples to offload, where we seek maximal accuracy of the classification and compare it with a simple random selection. Finally, we also investigate the option to minimize false negative rates as opposed to improving the accuracy of the entire classification, which can be useful for

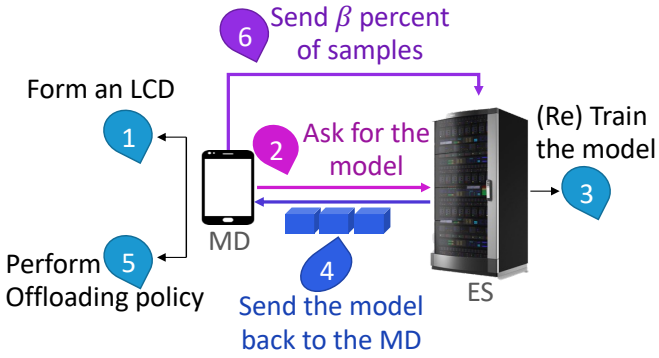


Fig. 1: System architecture of the proposed method.

certain specific applications. Importantly, the proposed domain-constrained method is different from existing methods in the literature, which mostly rely on a general domain of the dataset with model simplification. Instead, our technique focuses on local domains to build lightweight classifiers.

II. Preliminaries and System Architecture

The need for the timely analysis of real-time data characterizes a broad spectrum of applications. To make an example, ML approaches can be used for the prediction and early detection of emergencies in many IoT applications of assisted living and smart environments. This includes smart healthcare, autonomous driving, smart cities, and monitoring of industrial plants [13], [14], [15].

From an abstract perspective, such detection algorithms can be built through the collection of considerably large datasets tracking many parameters in a centralized repository, and the application of complex ML techniques. Delay and connectivity constraints push toward a paradigm where the ML algorithm is executed at the ES and part of them, albeit simplified, to the MDs themselves. A tradeoff arises between a rapid localized analysis of user data, but with lower accuracy, and a more refined analysis performed at the ES [16].

We focus on a general problem of binary classification on a set of supervised data samples where each data sample needs to be classified as belonging to either the 0 or 1 class. We note that, depending on the application, false negatives can be a source of larger concern compared to false positives as they can potentially lead to missing a due intervention [12]. For this reason, in the following analysis, we will explore both the case where errors are equivalent and a scenario where we emphasize the need to avoid false negatives.

Fig. 1 shows the system architecture, where the MD is responsible for collecting data and forming the LCDs, defined as subregions of the overall feature space. We assume that, while the ES is capable of accurately classifying all samples through sophisticated deep learning, the ML techniques employed at the MDs are much more limited. If the input sample distribution is known a priori, simpler classifiers (e.g., linear support vector machines) can be used by focusing on a specific feature region, obtaining good accuracy. Such knowledge of the distribution can be assumed by a continuous domain adaptation process performed by the ES, training (and retraining) the parameters of the local classifiers for each LCD. This is driven by a lightweight deep reinforcement learning (DRL) agent residing at the ES [3], [17], which controls resource allocation. This would allow conveying the rich knowledge available at the ES down to the MD by exchanging appropriately trained labeling parameters. Then, the MD performs the classification and employs an offloading policy to send a fraction of the samples to the ES.

The performance of local classification is highly sensitive to the size of the LCDs and the ability of the ES to update the model parameters in the system. In particular, finely tuned training of a large number of domains can achieve good accuracy but would be computationally heavy. If the domains are too small, the ES can have too few samples to properly estimate the correct parameters. On the other hand, decreasing the number of domains diminishes the effectiveness of the approach itself. For this reason, in the following, we investigate this tradeoff by varying the number of domains. At the same time, we will assume an idealized instantaneous training that is to be interpreted as an upper bound. However, as the main point of our investigation, we will show that both local classification and offloading techniques can be jointly employed, thereby leading to the high accuracy of the classification with overall limited and manageable efforts in terms of computation and communication exchanges.

III. System Model

Our two-tier edge-enabled system contains an IoT layer and an edge layer. The former consists of M MDs, where $i = 1, \dots M$ indexes the ith MD, whereas the latter includes an ES interconnected with the MDs through reliable wireless links, which are used to submit parameters from the ES for the local classification in the MDs, as well as to offload samples from the MDs to the ES. Since the capacity of the communication channel is limited, it ought to be used parsimoniously. The main objective of our investigation is to show that very high accuracy can be achieved by combining the two approaches of edge offloading and local classification, both with limited resource usage, as opposed to only using either of them, which would lead to achieving high accuracy only under an excessive usage of the wireless channel.

While the data collection operation is periodically repeated by every MD over time, we concentrate our attention on a specific time frame, where a generic MD acquires N samples represented by $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, each of them being a vector of n real values, i.e., $\mathbf{x}_j \in \mathbb{R}^n$ for all j. The objective of this architecture is to perform a binary classification by which each $\mathbf{x}_j$ is assigned to either a positive or a negative class. We let $\delta : \mathcal{X} \mapsto \{0, 1\}$ be the ground-truth function, according to which the set of all possible samples X can be partitioned into two disjoint subsets $\mathcal{X}_1$ and $\mathcal{X}_0 = \mathcal{X} \backslash \mathcal{X}_1$ corresponding to the

positive and negative classes, respectively, where $\delta(\mathbf{x}) = k$ iff $\mathbf{x} \in X_k$, $k \in \{0, 1\}$.

We assume that a perfect implementation of δ is available at the ES, for instance as a result of a careful training process that allows the acquisition of a near-perfect knowledge of the distribution of samples [18]. Instead, this is not accessible at the MDs due to their inherent limitations, therefore, they implement a simplified auxiliary decision rule $\tilde{\delta}_\Theta$, parameterized by Θ. That is, at each MD, $\tilde{\delta}_\Theta : X \mapsto \{0, 1\}$ is used to approximate the inaccessible ground-truth δ with $\tilde{\delta}_\Theta(\mathbf{x})$, based on the sample $\mathbf{x} \in X$. However, the computational complexity of δ and $\tilde{\delta}_\Theta$ are different, and $\tilde{\delta}_\Theta$ can be considered as a lossy form of δ [6]. Hence, the classification can be performed in the form of either local or external classification. The MD can also check first if the set of the model parameters, i.e., Θ, guarantees an accurate prediction for the sample, and exploit this assessment to decide whether to perform the classification locally or externally at the ES' side, where the label can be computed with δ that is a perfect match with the ground-truth.

This implementation can be further refined via some other details that for the sake of simplicity are omitted here, as they are out of scope with the present analysis that is just concerned with harmonizing local and external classifications to achieve the highest accuracy possible. First, we consider that Θ is always updated by the ES at each frame so that the model available at the MDs is closely matching as their local distribution change within the data and the only losses are due to the inefficiency of the domain specialization and the lower complexity of the local classifier. The auxiliary decision rule $\tilde{\delta}_\Theta$ would require supervision from the ES with the transmission of parameters Θ at a rate that depends on the dynamics of the data themselves. This is also another aspect that is out of the scope of the present evaluations, as we consider an instantaneous classification of the data, without any dynamics in their acquisition.

At any rate, the exchange of parameters Θ, and the resulting occupancy on the ES-MD channel, is proportional to the number of LCDs. Thus, it is interesting to evaluate the impact of the number of LCDs on the accuracy. Finally, it is worth observing that Θ represents a distribution-wise set of parameters for the entire dataset. Thus, it is more or less constant (and becomes more precise) if the dataset grows. On the other hand, the data offloaded to the ES grow proportionally with the size of the dataset to evaluate, as we consider a tunable *fraction* of the data to offload.

Accordingly, the independent parameters to explore in our architecture are (i) the fraction β of data offloaded to the ES; and (ii) the number n_c of LCDs in the local online domain-constrained classification. An efficient harmonization of our techniques is achieved if we can reach adequate accuracy in the classification for low values of both β and n_c. Moreover, it is worth exploring how the selection of the specific data to offload is performed, since the objective is to send to the ES the samples for which the classification is more likely to be incorrect, to exploit the better classification rule δ.

The model at the ES is trained with stochastic gradient descent using cross-entropy loss, defined as

$$f(\mathbf{X}, \Theta) = -\delta(\mathbf{X}) \ln \tilde{\delta}_\Theta(\mathbf{X}) - \big(1-\delta(\mathbf{X})\big) \ln\big(1-\tilde{\delta}_\Theta(\mathbf{X})\big). \quad (1)$$

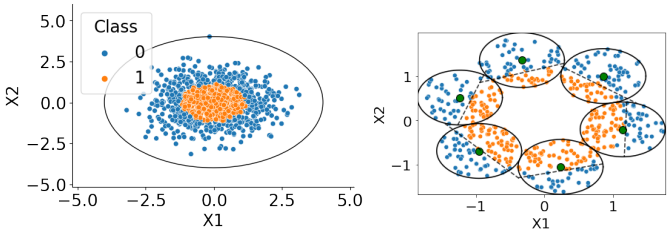
Using the logistic regression theory, the error probability is

$$P_{\text{err}}(\tilde{\delta}; \mathcal{P}) = 1 - \underbrace{\tilde{\delta}_\Theta(\mathbf{X})^{\delta(\mathbf{X})} \Big(1 - \tilde{\delta}_\Theta(\mathbf{X})\Big)^{(1-\delta(\mathbf{X}))}}_{e^{-f(\mathbf{X},\Theta)}}, \quad (2)$$

that implies that the loss $f(\mathbf{X}, \Theta)$ can be used as a guiding criterion for minimizing the probability of misclassification as they have the same monotonic character, i.e., $P_{\text{err}}(\tilde{\delta}; \mathcal{P})$ increases with $f(\mathbf{X}, \Theta)$.

We use (1) both as an evaluation metric, as well as a guiding parameter to select the samples to offload to the ES; namely, each MD, will be allowed to send a fraction β of the samples to the ES for an error-free classification, and this will be decided either randomly, or with a greedy choice based on putting all samples in decreasing order of loss and sending those who score more on this metric, up to filling the allowed offloading ratio β.

As a side evaluation, we also consider the minimization of false negative ratio (FNR) as a possible alternative objective. Indeed, this would consider the same approach of increasing the accuracy, mainly focusing on avoiding false negatives, which may be more dangerous in certain applications.



(a) (b)

Fig. 2: Original dataset (a) vs application of linear classifiers in local domains trained on specific subsets (b).

IV. Performance Evaluation

In this section, we assess the proposed architecture in a scenario where an MD connected to an ES through an ideal channel tries to classify data as shown in Fig. 2.

The whole dataset consists of $N{=}10^4$ samples of synthetic data, each with $n{=}2$ features following a Gaussian distribution with zero mean and unit standard deviation, using the `make_gaussian_quantiles` function predefined in Python. The positive class contains samples with a feature norm less than 1, while outliers are the negative class. The ES is assumed to know the ground-truth and can send training parameters for the n_c specialized domains on the circle border, to allow the application of a support vector machine (SVM) with a linear kernel at the MD, also implemented in Python [19]. The channel capacity is intentionally unconstrained but at the same time, we investigate the accuracy achievable with minimal data exchanges between the MD and the ES.

Under our assumptions, each MD is responsible for collecting data in every time-frame, forming LCDs, and classifying the

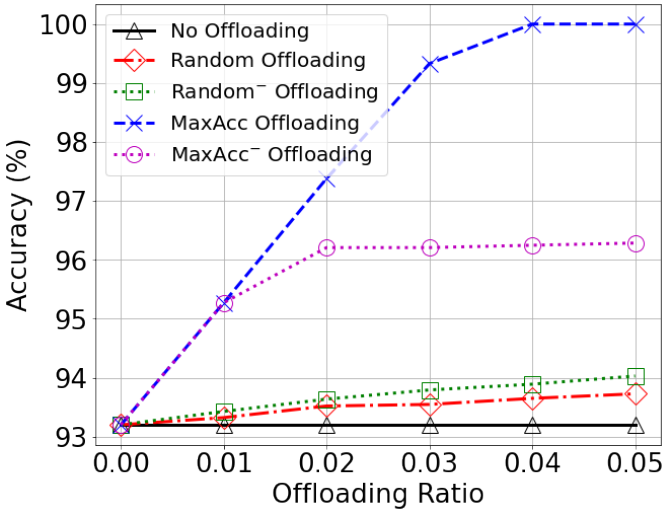


Fig. 3: Accuracy vs. β for $n_c = 4$ LCDs.

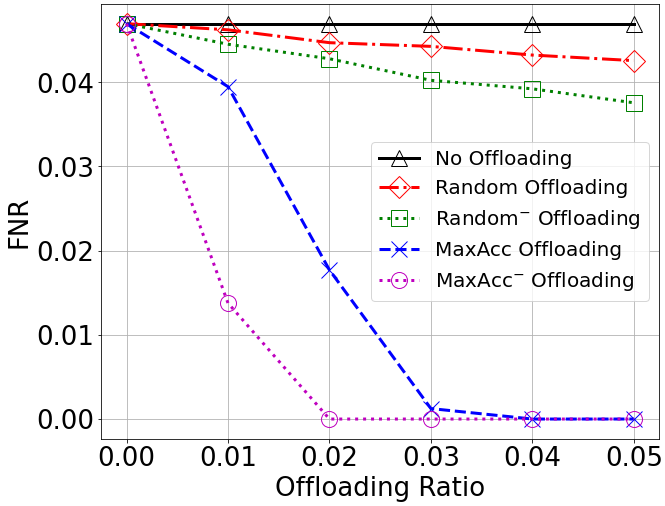


Fig. 5: False negative ratio vs. β for $n_c = 4$ LCDs.

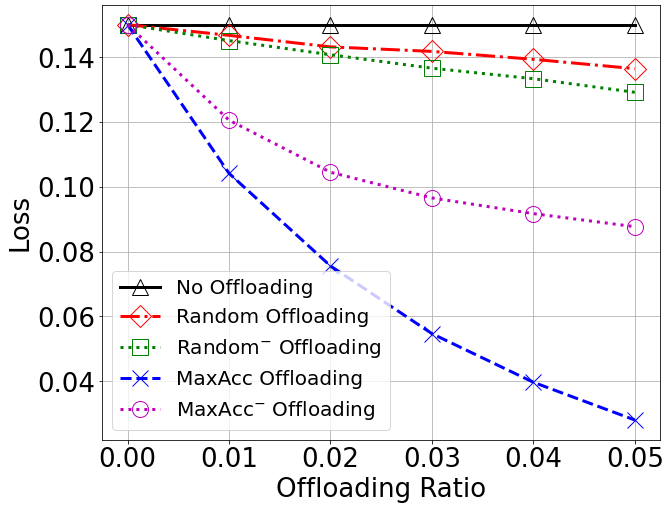


Fig. 4: Cross-entropy loss vs. β for $n_c = 4$ LCDs.

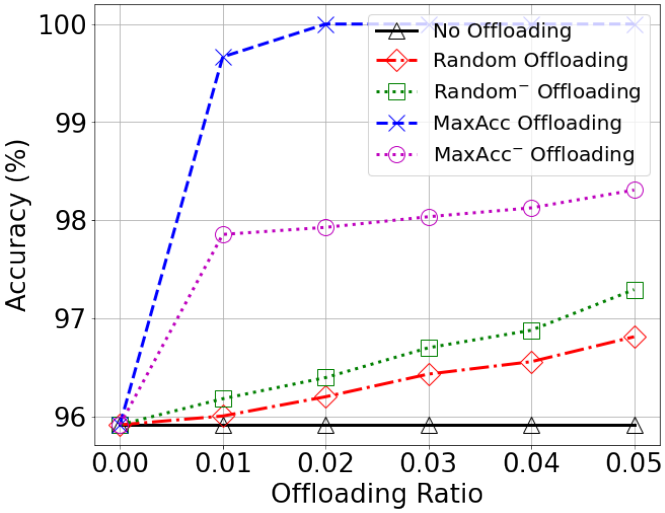


Fig. 6: Accuracy vs. β for $n_c = 6$ LCDs.

samples within each LCD, locally. The ES, on the other hand, is responsible for providing the MD with an updated model w.r.t every LCD the MD aims to perform the classification for. Finally, the MD can offload a fraction β of its data to the ES.

The performance of the proposed architecture is assessed in terms of accuracy, loss, and FNR, by changing the offloading ratio β and the number of LCDs n_c, under different policies for the selection of the samples to offload. The considered policies are listed below.

No Offloading: The model is updated for each LCD, yet no sample is offloaded to the ES for classification. This is our baseline performance to see the impact of applying LCDs.

Random Offloading: Used as a further benchmark, the model is updated for each LCD, and samples are randomly chosen to be offloaded to the ES.

Random$^-$ Offloading: The model is updated for each LCD and only samples with negative labels are randomly chosen to be offloaded, which is meant to decrease the FNR.

MaxAcc Offloading: The objective of this policy is to achieve accurate predictions by offloading dubious samples to the ES. To this end, the samples are sorted in descending order of prediction loss f; thus, the samples with the highest loss are offloaded to the ES.

MaxAcc$^-$ Offloading: This policy aims to maximize the accuracy of the negative predictions with the lowest accuracy to the ES. Similar to the previous one, samples are sorted in decreasing order of f, but only negative predictions are offloaded to the ES.

Two investigations are considered in this regard. In the first one, simulations are performed for $n_c \in \{4, 6, 8\}$ LCDs. For each n_c, the samples are classified for different values of the offloading ratio β. In the second one, we set $\beta \in \{0.02, 0.05\}$ and we consider a variable number n_c of LCDs.

Fig. 3 considers accuracy versus the offloading ratio β;

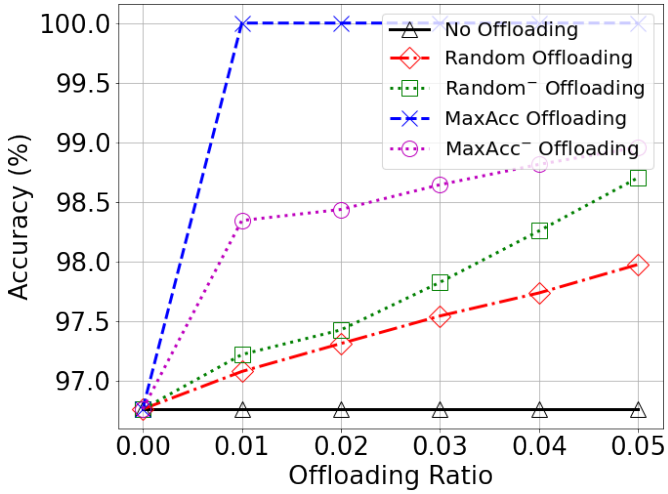


Fig. 7: Accuracy vs. β for $n_c = 8$ LCDs.

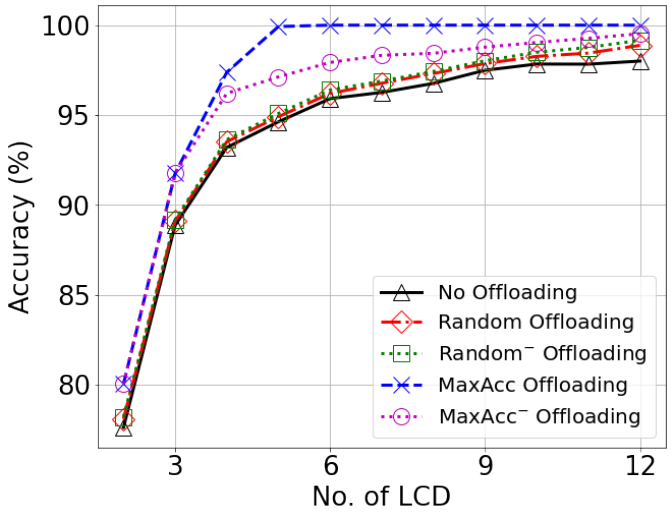


Fig. 8: Accuracy vs. n_c for offloading ratio β = 0.02.

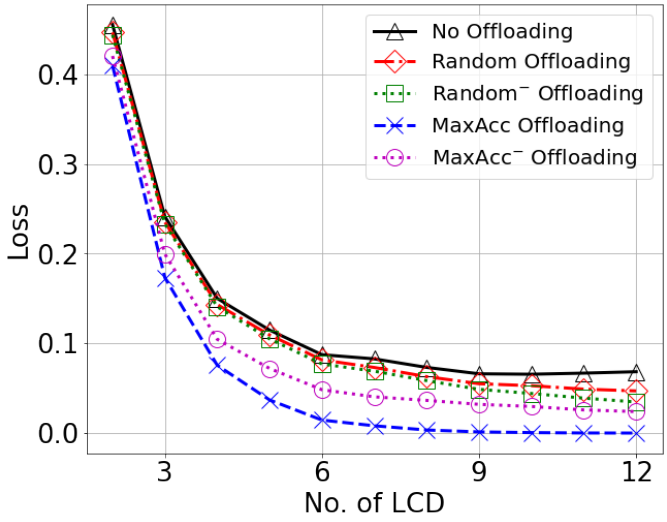


Fig. 9: Cross entropy loss vs. n_c for offloading ratio β=0.02.

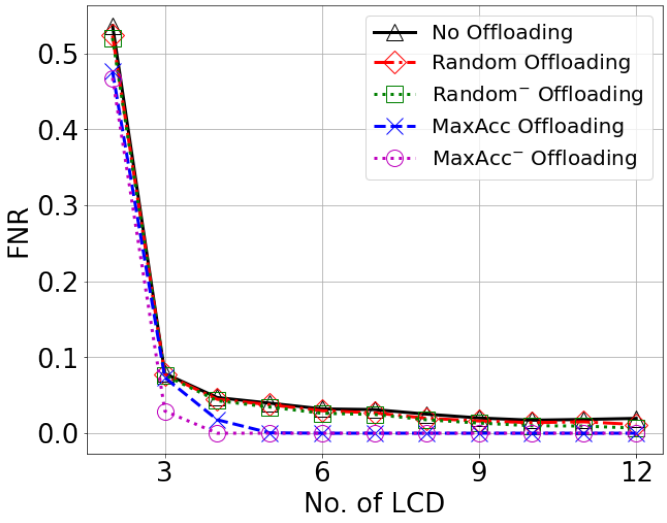


Fig. 10: False negative ratio vs. n_c for offloading ratio β=0.02.

without any offloading, the accuracy is just 93%, whereas MaxAcc Offloading converges to an accuracy of 100% with $\beta \geqslant 4\%$. The Random Offloading policies do not significantly improve the accuracy, thus proving the importance of a careful selection of samples offloaded to the ES.

Fig. 4 confirms this trend from the point of view of the loss, which is the opposite of the accuracy, as per (2). Fig. 5 instead shows that MaxAcc$^-$ Offloading is effective to decrease the FNR. Once again, some offloading is required but $\beta \geqslant 0.02$ is enough to push the FNR to zero.

Figs. 6 and 7 indicate instead the accuracy for a higher number of LCDs, equal to 6 and 8, respectively. Loss and FNR are not shown for the sake of brevity, yet they have the same trend as Figs. 4 and 5. These plots demonstrate how a higher number of LCDs improve the accuracy as the distribution of samples is more closely matched. Still, some offloading is required to get a fully accurate classification. Clearly, one can trade off β for n_c but these results strongly justify our proposal of harmonizing local domain classification and offloading to the ES.

To show the effect of n_c on the performance of the proposed policies, we perform simulations for different values of n_c with respect to constant β. Results obtained in Figs. 3, 6, and 7 indicate that $\beta = 0.05$ always converges the accuracy of 100%. On the other hand, the accuracy is variable between 97% to 100% for $\beta = 0.02$. Figs. 8–10 show accuracy, loss, and FNR, respectively, for different values of n_c and $\beta = 0.02$. In general, as seen in Figs. 8–9, by increasing n_c, the performance is improved.[1]

Figs. 8 and 9 show the superiority of MaxAcc Offloading over the other policies, whereas Fig. 10 implies that MaxAcc$^-$

[1]This improvement works only from a theoretical standpoint, because, in practical scenarios, if the size of an LCD is tiny, the ES can have problems in estimating the model parameters Θ.

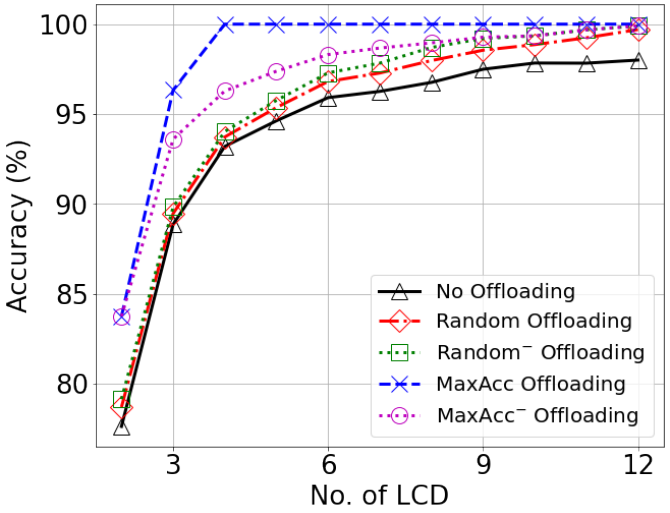


Fig. 11: Accuracy vs. n_c for offloading ratio β = 0.05.

Offloading can be very effective in reducing the FNR. All of these results were achieved for limited offloading of $\beta = 0.02$, and Fig. 11 confirms these results for $\beta = 0.05$.[2] Overall, this shows that near-perfect accuracy of classification (as well as low cross-entropy loss and false negatives) can be achieved by carefully combining offloading and domain classification, with just limited effort from either component.

V. Conclusions

We investigated a two-tier architecture for data classification in IoT scenarios, where both individual devices and the reference ES have decision capabilities, but with different levels of accuracy. We proposed a harmonized approach that combines offloading to the ES of the most critical data, while at the same time adopting a domain classification on the majority of the data for local processing at the MDs. We analyzed the resulting performance and quantitatively showed that the two proposed policies are able to complement each other, therefore leading to a near-100% accuracy with limited offloading and a reasonable choice of the local classification domains. This validates the option of such an architecture for efficient decision-making in data-intensive contexts.

Future developments include the analysis and implementation of this technique in practical applications with real data, possibly involving dynamic data acquisition in transient phases. At the same time, it would be interesting to frame the analysis by accounting for the constraints in terms of computational complexity and communication capabilities of the ES-MD links with a best-effort approach, as well as a real-time choice of the harmonization parameters between the two techniques in a fully adaptive setup without any prior knowledge of data distribution or their size.

[2]Loss and FNR diagrams have not been provided to avoid redundancy.

References

[1] S. Zhou, L. Wang, S. Zhang, Z. Wang, and W. Zhu, "Active gradual domain adaptation: Dataset and approach," *IEEE Trans. Multimedia*, vol. 24, pp. 1210–1220, 2022.

[2] R. Hadidi, J. Cao, M. S. Ryoo, and H. Kim, "Toward collaborative inferencing of deep neural networks on internet-of-things devices," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4950–4960, 2020.

[3] Y. Matsubara, R. Yang, M. Levorato, and S. Mandt, "Supervised compression for resource-constrained edge computing systems," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, 2022, pp. 2685–2695.

[4] Y. Yan and Q. Pei, "A robust deep-neural-network-based compressed model for mobile device assisted by edge server," *IEEE Access*, vol. 7, pp. 179 104–179 117, 2019.

[5] R. El Shawi, Y. Sherif, M. Al-Mallah, and S. Sakr, "Interpretability in healthcare a comparative study of local machine learning interpretability techniques," in *Proc. IEEE CBMS*, 2019, pp. 275–280.

[6] J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, and P. Mohapatra, "Edge cloud offloading algorithms: Issues, methods, and perspectives," *ACM Comp. Surv.*, vol. 52, no. 1, pp. 1–23, 2019.

[7] A. V. Guglielmi, M. Levorato, and L. Badia, "A Bayesian game theoretic approach to task offloading in edge and cloud computing," in *Proc. IEEE ICC Wkshps*, 2018, pp. 1–6.

[8] F. S. Abkenar, K. S. Khan, and A. Jamalipour, "Smart-cluster-based distributed caching for fog-iot networks," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3875–3884, 2020.

[9] F. Sufyan and A. Banerjee, "Computation offloading for distributed mobile edge computing network: A multiobjective approach," *IEEE Access*, vol. 8, pp. 149 915–149 930, 2020.

[10] X. Li, W. Zhang, Q. Ding, and J.-Q. Sun, "Multi-layer domain adaptation method for rolling bearing fault diagnosis," *Sign. Proc.*, vol. 157, pp. 180–197, 2019.

[11] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, 2019.

[12] A. S. Assiri, S. Nazir, and S. A. Velastin, "Breast tumor classification using an ensemble machine learning method," *J. Imag.*, vol. 6, no. 6, p. 39, 2020.

[13] I. Azimi, A. Anzanpour, A. M. Rahmani, T. Pahikkala, M. Levorato, P. Liljeberg, and N. Dutt, "HiCH: Hierarchical fog-assisted computing architecture for healthcare IoT," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 5s, pp. 1–20, 2017.

[14] L. Prospero, R. Costa, and L. Badia, "Resource sharing in the internet of things and selfish behaviors of the agents," *IEEE Trans. Circuits Syst. II*, vol. 68, no. 12, pp. 3488–3492, 2021.

[15] P. Ambika, "Machine learning and deep learning algorithms on the industrial internet of things (IIoT)," *Adv. Comput.*, vol. 117, no. 1, pp. 321–338, 2020.

[16] V. Mancuso, P. Castagno, M. Sereno, and M. A. Marsan, "Stateful versus stateless selection of edge or cloud servers under latency constraints," in *Proc. IEEE WoWMoM*, 2022, pp. 110–119.

[17] S. Chen, Z. Yao, X. Jiang, J. Yang, and L. Hanzo, "Multi-agent deep reinforcement learning-based cooperative edge caching for ultra-dense next-generation networks," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2441–2456, 2020.

[18] E. Recayte and A. Munari, "Caching at the edge: Outage probability," in *Proc. IEEE WCNC*, 2021, pp. 1–6.

[19] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learning Res.*, vol. 12, pp. 2825–2830, 2011.