Title	Distributed classification with dynamic communication for air quality sensing
Authors	Nash, Andrew;Pesch, Dirk;Guha, Krishnendu
Publication date	2025-08-27
Original Citation	Nash, A., Pesch, D. and Guha, K. (2025) 'Distributed classification with dynamic communication for air quality sensing', IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Kalamata, Greece, 06-09 July 2025, pp. 1–6. https://doi.org/10.1109/ISVLSI65124.2025.11130290
Type of publication	Conference item
Link to publisher's version	https://ieeexplore.ieee.org/xpl/conhome/11129697/proceeding - 10.1109/ISVLSI65124.2025.11130290
Rights	© 2025, IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-08-28 22:07:53
Item downloaded from	https://hdl.handle.net/10468/18268

Distributed Classification with Dynamic Communication for Air Quality Sensing

Andrew Nash
School of Computer Science and IT
University College Cork
Cork, Ireland
anash@umail.ucc.ie

Dirk Pesch
School of Computer Science and IT
University College Cork
Cork, Ireland
dirk.pesch@ucc.ie

Krishnendu Guha
School of Computer Science and IT
University College Cork
Cork, Ireland
kguha@ucc.ie

***Abstract*—A body of work has focused on distributed task classification in sensor networks. The majority of works in this domain focus on either decision fusion, edge/fog computing (or equivalently, data fusion), split computing (feature fusion) or fully distributed agents. These methodologies, combined with machine learning classifiers, are being applied to problems such as UAV and robot guidance, smart cities and smart grids [1]. In contrast, we propose and define a novel hybrid strategy that makes greedy local decisions at end devices as to whether to operate in a distributed or split computing context. We devise a communication strategy for distributed classification, which balances communication efficiency with classification accuracy. We demonstrate our proposed methodology on a real-world IoT dataset that consists of air-quality sensor readings taken in a home environment. Experimental results depict that our proposed approach balances communication cost and classification accuracy between known optimal values. In addition to this, our proposed strategy can achieve high accuracy at lower cost than traditional methods. We explore in detail the conditions under which our model performs well, and determine some potentially promising avenues for future extension and development.**

***Index Terms*—Distributed processing, split computing, edge computing, IoT**

I. Introduction

Internet of Things (IoT) platforms are increasingly generating large amounts of distributed data, leading to increased importance of scaleable, secure and efficient data processing and distributed intelligence functionality [1] [2] [3].

Conventionally, research for optimizing communication efficiency of joint classification tasks over distributed sensor networks have applied principles of either edge/cloud computing [4] over potentially hierarchical servers [5], or fully-distributed collaboration [6]. We expect that by devising a method for end devices to dynamically choose whether to employ fully distributed collaborative or edge-computing based modes of operation, we can trade off and optimize communication costs and classification accuracy.

To demonstrate the effectiveness of our proposed strategy, we utilise an air quality dataset [7] that focuses on the detection and classification of various household activities that affect indoor air quality based on readings from a heterogeneous set of gas sensors. The data was collected from a small and controllable environment that allows us to explore the application of our proposed scheme of classification against comparable conventional classification schemes.

Our key contributions in this paper can be summarized as:

- Developing a dynamic communication strategy that performs a joint classification task, which
  1) balances between data offloading to edge servers with requests for data from nearby sensor devices,
  2) optimizes communication costs,
  3) maintains high classification accuracy
- Demonstrating the application of this proposed communication strategy on a real-world dataset, using simple classifiers on a model of an ad-hoc wireless network
- Comparing the classification accuracy and corresponding communication cost under our proposed strategy against the following baseline distributed classification methods
  1) fully independent classifiers on each sensor
  2) fully distributed classifiers, with some inter-communication between sensors
  3) a centralised edge classifier over all data

This paper is organised as follows. Section II discusses relevant literature and motivation. The proposed methodology is presented in Section III. Section IV details experimental setup and result analysis. Section V concludes the paper with a discussion on future work.

II. Background and Motivation

A. Related Works

In this section, we highlight relevant research on efficient distributed inference across networks of sensor-equipped devices, with and without the use of edge computing. This research can be broadly categorised into **Edge computing/Data Fusion**, where a joint classifier on an edge server processes all data from sensor devices [4], **Split Computing** which partitions inference between the sensors and the edge [8], and **Fully Distributed** inference, where methods such as Bayesian networks or reinforcement learning are used to orchestrate data exchange between sensors. Decision fusion is often applied to the resulting predictions, to form a consensus [9].

Table I presents a taxonomy of classification and communication schemes in the literature. We also depict the aspects in which our proposed mechanism fares better than the existing schemes. We identify methods such as [5] as having a *dynamic computation graph* – meaning that during the inference process,

This publication has emanated from research conducted with the financial support of Taighde Éireann – Research Ireland under Grant number 18/CRT/6222 (ADVANCE CRT) and Grant number 13/RC/2077_P2 (CONNECT: the Research Ireland Centre for Future Networks)

the model makes dynamic decisions to integrate various data sources. Separately, some methods perform inference over a fixed communication scheme, where the path through which all data is transferred is constant and data-independent. We refer to methods that dynamically transfer data to different locations (regardless of whether classification is performed on a constant set of data sources or not) as having a *dynamic communication graph*. In works such as [10], where the communication graph is dynamic, Computation Resource Allocation techniques are used to transmit data based on minimum expected cost (by a pre-defined metric) of inference depending on the overall state of the network.

Table I: Communication & Inference Scheme Comparison

Reference	Edge Server	Distributed	Dynamic Comp.	Dynamic Comm.
[8][6]	✓	-	-	-
[5]	✓	-	-	✓
[11]	✓	-	✓	-
[10] [12]	✓	✓	-	✓
[9][13][14]	-	✓	-	✓
Proposed	✓	✓	✓	✓

B. Motivation

We consider the following situations:

1) Local observations at each end device are sufficient to make a confident classification;
2) Clusters of observations from small groups of end devices are sufficient to make a confident classification;
3) Neither individual nor clusters of local observations at each end device are sufficient to make a confident classification, but a joint classifier can make a confident classification over all of the data;

As we see in Table I, current techniques capture at most two of these three situations with any given system.

In [1], the authors have conducted a comprehensive review of the state-of-the art in distributed Artificial Intelligence. They discuss the concept of *Communication-Efficient Multi-Agent Reinforcement Learning*, which corresponds to distributed inference in Section II-A. They identify the heterogeneity of the decision-making agents as an important issue requiring further research. Hence, we cannot simply consider edge servers as agents alongside end-device agents in a traditional distributed classification scheme.

To the best of our knowledge, there are no works in the literature where our concepts of both dynamic *computation* and *communication* graphs are jointly applied in networks with communicating end devices, and a centralized edge server. It is our objective to optimize communication efficiency over existing methods by devising decision-making modules that exist on end-devices and select between local classification, distributed classification between end devices and classification on a centralised edge server.

III. Proposed Methodology

A. Environmental & Network Conditions

We consider a simplified model of the conditions under which the data in [7] were collected. From this we assume:

- Communication was through an infrastructure-mode 2.4GHz WiFi network. To enable potential low-power inter-sensor communication, we assume that communication takes place via direct ad-hoc connections.
- There exists a central edge-like device (analogous to the ThingSpeak platform in the original research), equidistant from all sensors.
- Final detections need to be delivered to a mobile device in the room. This device is only permitted to perform decision fusion of multiple predictions

We simulated various network topologies and, to each we apply various classification methods to the dataset.

In our simulation, we assumed that there are bi-directional ad-hoc wireless channels available between randomised pairs of sensors, such that each sensor can communicate directly with a transmission power of T_s mW with some set of other sensors. Each sensor can communicate with the edge server at a transmission power of T_e mW. We further assume that reporting a classification outcome from a sensor requires T_s mW, and T_e mW from the edge server.

B. Data Assumptions

The purpose of this research is to demonstrate the potential utility of our novel dynamic inference strategy for low power sensor devices in the considered dataset. For this reason, we do not consider high-performance or complex classifiers.

The edge server will perform classification with K-Nearest Neighbours and the sensor devices perform classification using Naïve Bayes classifiers. This reflects the practical reality that the edge server will typically have higher storage and computation capability than the sensor devices.

1) Activity Class Probabilities: We assume that each of the the activity classes (Normal, Cleaning, Cooking, Smoke) are time-independent, and have a multinomial distribution.

2) Normality and Independence: We assume that the distribution of readings from each sensor s, conditioned on the activity a, has a Gaussian distribution with mean μ_{sa} and standard deviation σ_{sa}^2. We also assume that the sensor readings are conditionally independent given the underlying activity. This allows us to easily compute the terms in our decision equations, illustrated in Fig. 2. Tests of these assumptions were performed. As the MG-811 sensor's data appears to violate our Normality assumptions (Fig 1), and empirical experiments showed it does not lead to an increase in performance, it is excluded from our experiments.

C. Baseline Classification Models

1) A joint classifier on the edge server: implemented by a K-NN classifier with $K = 3$ to coincide with the original solution proposed in [7]. We do not consider more complex classifiers here, since K-NN attains high accuracy on this dataset, and further, the objective of this research is not to attain maximal possible classification accuracy. Our focus is to devise the communication *communication strategies* that balance the accuracy performance of a given classification model on the edge, and the communication efficiency of distributed classifiers.

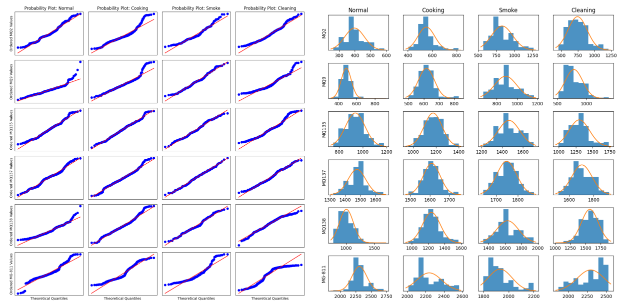

Figure 1: MG-811 corresponds to the bottom row of the QQ plots (left), and the distribution plots (right). Columns correspond to activity classes. Orange lines correspond to bootstrap fitted Gaussian distributions.

2) A series of independent classifiers: which all report to the mobile device which performs decision fusion:

$$\underset{a}{\operatorname{argmax}}\, P\left(a|s_i\right) = \frac{f\left(s_i|A=a\right)p_a}{\sum_{j=1}^{F} f\left(s_i|A=j\right)p_j}$$

3) A fully distributed structure: where each sensor can directly report to the mobile device if it has sufficient confidence ($p > \lambda$), or otherwise request data from a single connected sensor of its choice.

$$\begin{cases} \underset{a}{\operatorname{argmax}}\, P\left(a|s_i\right), \text{if } \underset{a}{\max}\, P\left(a|s_i\right) > \lambda \\ \underset{a,j}{\operatorname{argmax}}\, P\left(a|s_i, s_j\right) \text{ s.t. } j = \underset{a,j}{\operatorname{argmax}}\, E_j[P\left(a|s_i, s_j\right)] \end{cases}$$

D. Proposed Communication & Classification Method

Our proposed communication strategy is effectively a hybridization of the baselines defined above, composed of three components:

1) Where there is sufficient confidence in a sensor's data, a report is directly made to the mobile device.
2) Otherwise, we compute the expected confidence $\hat{p}_{i,j}$ of predictions made jointly with this sensor's data and each available peer sensor. If the best expected confidence is sufficiently high, data is requested from its corresponding sensor. Specifically, we compute the expected cost of a request $2T_s\hat{p}_{i,j} + (T_s + 2T_e)(1 - \hat{p}_{i,j})$. If this is less than $2T_e$, (expected cost an edge offload), we perform the request. If the joint prediction is sufficiently high, the result is sent directly to the mobile device.
3) If neither of the preceding steps results in a sufficiently confident prediction, it is necessary to offload this sensor's data to the edge device, which will report its prediction to the mobile device.

We assume that each of the end devices executes the steps above in parallel. At each time-step, we assume that the edge server routinely performs joint inference over all data-points reported at that time-step. This classification is performed by 3-NN, as per the baseline edge classifier. The edge is assumed to reports its classified activity to the mobile device with cost T_e. The mobile device performs decision fusion between the reported activities from the edge server and sensors. The edge

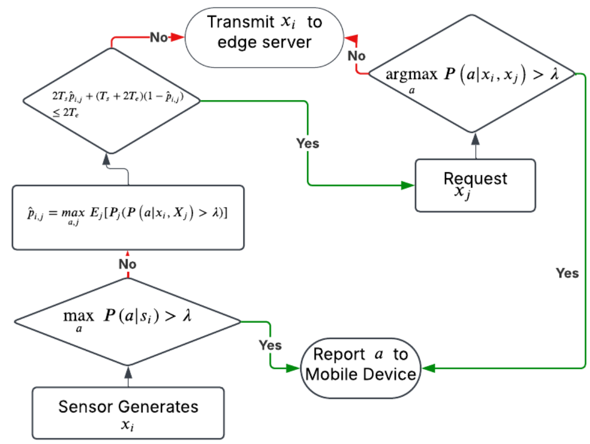

Figure 2: Process-flow of our proposed scheme (III-D)

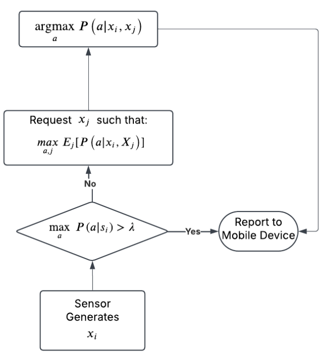

Figure 3: Communicating-distributed baseline (III-C)

server's prediction is weighted by the number of sensors whose data was used to generate it.

In both our proposed model (detailed equations in Fig. 2), and the distributed baseline (III-C3), λ is the user-defined threshold used to determine when a predicted probability is sufficiently high to lead to a direct report to the mobile device.

E. Metrics & Optimization

For each sensor reading, we approximate the total energy expended on communication. We assume the power consumption for classification of each data-point is the sum of all transmission powers along each link of the transmission path from its sensor to the target mobile device. This is assumes that the data being transmitted are all uniform in size. In complex split computing scenarios, where variable-sized data are transmitted between devices (raw, processed or compressed by some means) we could extend this model by weighting by the message size at each step (and potentially also by signal path loss, interference factors, etc.). For our proposed method, we illustrate the trade-off between accuracy and communication cost (by the aforementioned metric of communication power) for various values of λ.

F. Validation & Parameter Fitting

We perform 10-fold cross validation, with the same split for each experiment. Within each experiment we estimate μ_{sa}, σ^2_{sa} from 9999 bootstrap samples on each training fold.

Table II: Results under edge baseline (III-C1)

Comm. Cost mW	Accuracy	Parameters T_s	T_e
12.0	0.96	1	2
30.0	0.95	1	5
60.0	0.95	1	10
90.0	0.95	1	15

Table III: Results under distributed baseline (III-C3)

Comm. Cost mW	Accuracy	Rep^d	Experimental Parameters T_s	T_e	Network	λ
7.25	0.76	0.55	1	2	Fully C.	0.7
7.25	0.76	0.55	1	15	Fully C.	0.7
7.25	0.75	0.55	1	2	Pairs	0.7
7.25	0.75	0.55	1	15	Pairs	0.7
7.25	0.78	0.55	1	2	Random	0.7
7.25	0.78	0.55	1	15	Random	0.7
7.25	0.78	0.55	1	2	Cyclic	0.7
7.25	0.78	0.55	1	15	Cyclic	0.7

Rep^d denotes the proportion of cases where a sensor can directly perform inference.

Table IV: Results under our proposed method (III-D)

Comm mW	Acc.	Rep^d	Rep^p	Experimental Parameters T_s	T_e	Network	λ
7.64	0.79	0.79	(0.09,0.73)	1	2	Fully C.	0.6
10.01	0.87	0.55	(0.11,0.64)	1	2	Fully C.	0.7
9.09	0.77	0.79	(0.18,0.7)	1	5	Fully C.	0.6
16.16	0.84	0.55	(0.31,0.53)	1	5	Fully C.	0.7
11.82	0.77	0.79	(0.19,0.7)	1	10	Fully C.	0.6
24.23	0.82	0.55	(0.38,0.53)	1	10	Fully C.	0.7
14.48	0.77	0.79	(0.2,0.71)	1	15	Fully C.	0.6
31.65	0.82	0.55	(0.4,0.54)	1	15	Fully C.	0.7
7.74	0.79	0.79	(0.08,0.72)	1	2	Pairs	0.6
10.09	0.87	0.55	(0.09,0.57)	1	2	Pairs	0.7
9.73	0.77	0.79	(0.17,0.66)	1	5	Pairs	0.6
16.73	0.84	0.55	(0.27,0.54)	1	5	Pairs	0.7
12.84	0.77	0.79	(0.18,0.67)	1	10	Pairs	0.6
25.51	0.82	0.55	(0.33,0.55)	1	10	Pairs	0.7
15.92	0.77	0.79	(0.19,0.67)	1	15	Pairs	0.6
33.7	0.82	0.55	(0.36,0.56)	1	15	Pairs	0.7
7.74	0.79	0.79	(0.06,0.73)	1	2	Random	0.6
9.68	0.88	0.55	(0.06,0.64)	1	2	Random	0.7
10.02	0.78	0.79	(0.14,0.73)	1	5	Random	0.6
16.74	0.85	0.55	(0.21,0.56)	1	5	Random	0.7
13.84	0.78	0.79	(0.15,0.74)	1	10	Random	0.6
27.11	0.85	0.55	(0.26,0.55)	1	10	Random	0.7
17.56	0.78	0.79	(0.16,0.74)	1	15	Random	0.6
36.68	0.85	0.55	(0.29,0.56)	1	15	Random	0.7
7.77	0.8	0.79	(0.06,0.68)	1	2	Cyclic	0.6
9.81	0.88	0.55	(0.07,0.77)	1	2	Cyclic	0.7
10.3	0.79	0.79	(0.15,0.64)	1	5	Cyclic	0.6
16.77	0.85	0.55	(0.23,0.61)	1	5	Cyclic	0.7
14.15	0.79	0.79	(0.16,0.66)	1	10	Cyclic	0.6
26.31	0.85	0.55	(0.29,0.58)	1	10	Cyclic	0.7
18.0	0.79	0.79	(0.17,0.67)	1	15	Cyclic	0.6
35.31	0.85	0.55	(0.31,0.59)	1	15	Cyclic	0.7

Rep^p denotes (p_1, p_2) - the proportion of cases where a sensor requests another sensor's data, and the proportion of these which lead to a direct report, respectively.

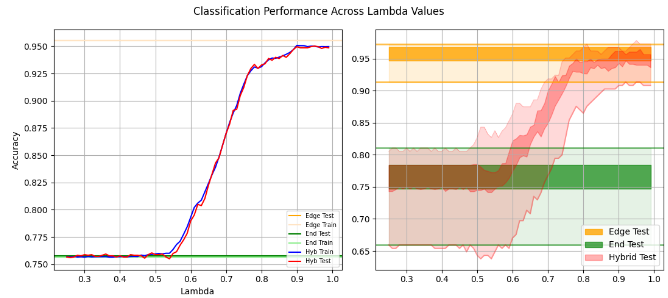

Figure 4: Classification accuracy of our scheme compared to end (III-C2) and edge (III-C1), with $T_s = 1, T_e = 2$ under the fully-connected network. Solid lines (left panel) indicate the mean across all validation folds. Lightly shaded areas indicate the max and min values across all folds, and dark shaded areas (right panel) the 25$^{\text{th}}$ and 75$^{\text{th}}$ percentiles.

IV. Experimentation & Results

Baseline and proposed classifiers were implemented in Python using NumPy and SciPy libraries.

A. Simulated Network Conditions

Four separate network architectures were evaluated (generated randomly using random seed 42 in NetworkX). Details of the generated graphs are included in the code. Every sensor is connected to the mobile device directly via ad-hoc connections with transmission power of T_s mW, and the edge device with a transmission power of T_e mW.

- **Fully Connected**: all sensors can request from one another.
- **Pairs**: each sensor has 2 to 3 neighbours from which it can request data, or have its data requested from.
- **Random**: each sensor has 1 to 3 neighbours from which it can request data, or have its data requested from.
- **Cyclic**: the sensors are arranged in a directional ring - each sensor can request data from a specific sensor, and will only have its data requested by a specific sensor.

B. Experimental Parameters

Experiments were performed with $T_s = 1$ and $T_e \in \{2, 5, 10, 15\}$. Our proposed model and the communicating-distributed baseline were evaluated for λ between 0.25 and 0.99 (inclusive) in increments of 0.01. All metrics are computed as the mean over the validation folds of each experiment.

C. Discussion of Experimental Results

Our objective is to develop a communication strategy for distributed classification that balances communication efficiency with classification accuracy. Our results on the chosen exemplar dataset indicate very stable performance over different network configurations and demonstrate that the choice of λ creates a trade-off between low communication cost and low classification accuracy and, in most cases, allows us to reach equivalent performance to the edge classifier at lower cost.

1) Comparison to Original Classification Accuracy: In the original dataset [7], Gambi et al. state that they achieved a classification accuracy of 96.32% on the multi-class classification task using a classifier that corresponds exactly to the edge baseline in our experimental configuration. This baseline

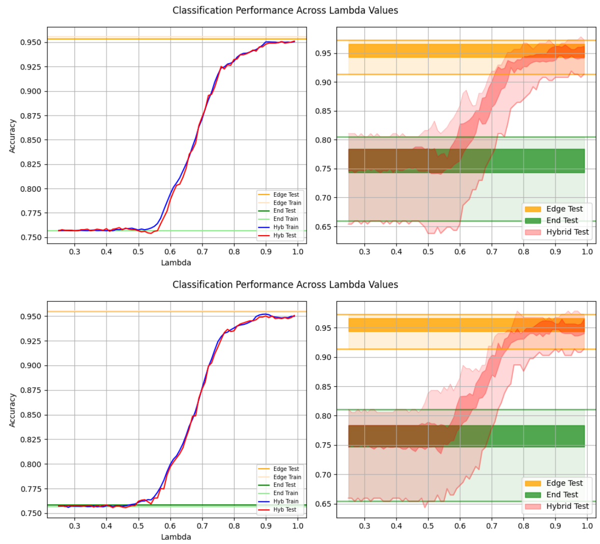

Figure 5: Compared to Fig. 4, classification performance remains stable in the Pair network (top), and the Cyclic network (bottom). These network graphs have no edges in common.

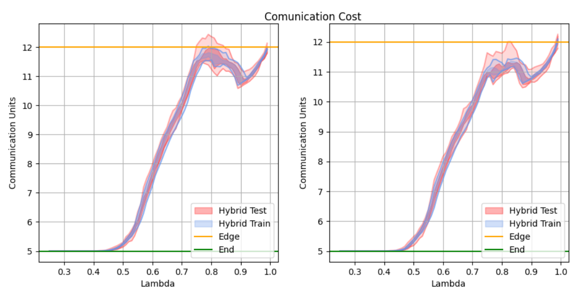

Figure 6: For a fully-connected network Fig. 4 demonstrates that choice of λ defines an accuracy tradeoff between baseline distributed and edge classifiers. Corresponding communication cost (left) shows a seemingly proportional tradeoff. Consistent results are seen under the cyclic network (right).

reproduces this result approximately, as expected – Table II and Fig. 4. Across various network conditions, our proposed scheme reaches this accuracy in both cross-validation training and test accuracy scores for high values of λ. With small values of T_e, this appears to occur for lower values of λ – compare Fig. 5 ($T_e = 2$) and Fig. 7 ($T_e = 10$).

2) Accuracy-Communication Cost Trade-off: The main region over λ of interest is where we achieve classification accuracy and communication cost between those of the baselines – since our classifier simply reduces to complete equivalence to the baseline classifiers at the extrema. Table IV gives an insight into the behaviour of our proposed strategy within this range. For $\lambda \in \{0.6, 0.7\}$ we can observe that all components of our strategy are in operation, with direct classification from sensors in $> 50\%$ of cases. Accuracy remains broadly stable for the same λ in each network as T_e increases. It is expected that the number of requests for other sensors' data might not be common, but should still occur in a noticeable number of instances - the region of the data meeting the criteria in

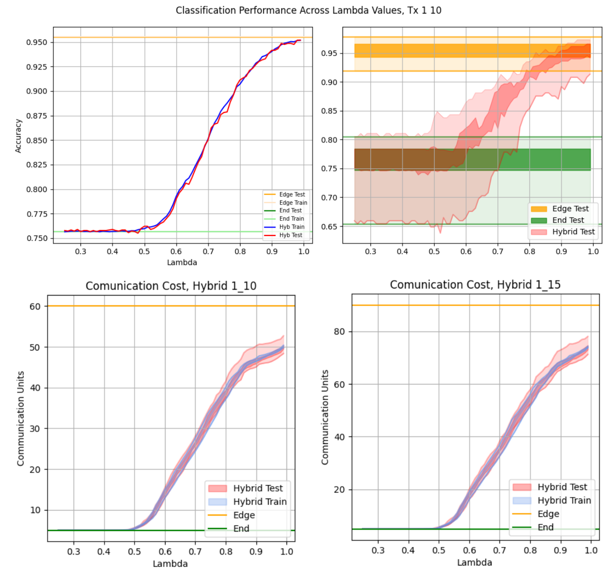

Figure 7: Our strategy gives accuracy under the $T_e = 10$ cyclic network (top) similar to that of the other networks. The communication cost outperforms the edge baseline at all values of $\lambda < 1$ when $T_e = 10$, and $T_e = 15$ (bottom left and right).

III-D, is not likely to be very large. In practice, we observe the percentage of instances where this occurs is highly variable between 2% and 38%. We notice that the proportion of cases where requests take place increases with T_e, as we would expect. We also observe that the percentage of these cases that result in a classification verdict being reached tends to be quite high. This is a good indication that all components of our strategy are being successfully exploited here.

3) Network Constraints: Overall classification performance of our proposed method remains stable across all four network architectures. In the extreme cases, the results can be seen to be stable between the fully connected architecture, whose results are depicted in Fig. 4), and the Cyclic architecture, whose results are depicted in Fig. 5.

4) Out-performance of End & Distributed Baselines: The communication cost for the end-classifier baseline (III-C2) was 5mW with an accuracy of 76% under all configurations. The results of the communicating-distributed baseline (III-C3) achieved marginal accuracy improvements – Table III, at slightly higher communication cost (total of 7.25mW).

While our proposed strategy has much higher communication costs than the distributed baselines for values of λ approximately > 0.5, this comes with higher accuracy – Fig. 4, where the orange and green lines depict the edge and independent-classifier baselines respectively.

5) Local Minima & Out-performance of Edge Baseline: In the tightly constrained cyclic network, there are indications in Fig. 5, and Fig. 6 that λ might not necessarily define a straight trade-off between the accuracy of distributed and edge baselines. Around $\lambda = 0.9$, we see local minima in validation training and test communication cost. The corresponding classification accuracy appears very close to the edge baseline in Fig. 5. Further, with increasing disparity between T_e and T_s, we observe even more distinct improvement of our proposed strategy over the edge baseline. For $T_e \in \{10, 15\}$, our proposed strategy has a lower communication cost for all values of $\lambda < 1$, at high accuracy level – Fig. 7

D. Utility of Our Proposed Methods

1) Network Parameters: These appear to have little impact on model performance, even between networks whose graphs are non-intersecting – Fig. 5. Utility of this proposed strategy on this dataset, can be reasonably assumed to not derive from our arbitrary choice of network graph, but from usefully capturing inter-dependencies in the data. Further, where edge offloads are relatively expensive/constrained in comparison to inter-sensor connections, we expect that our strategy can lead to noticeably lower cost classification with equivalent accuracy – Fig. 7.

2) Choice of λ: While this is highly data dependent, λ appears to behave as a trade-off parameter between communication efficiency and classification accuracy. In practice, this allows for prioritization of communication resources for critical tasks in a constrained network. We observe evidence across all configurations that our proposed strategy can reach similar classification accuracy as the edge baseline, at lower or at-worst equal cost. We expect that as this system scales, with sophisticated classifiers on the end devices, this effect will become more pronounced. This suggests that at larger scale this scheme should reach an equivalent level of classification performance as the edge server, at a much lower overall communication cost. We hypothesise that there is a regularizing effect of this method of having multiple classifiers which is contributing to this result. If so, this could also lead to not only accurate, but also more reliable and resilient classification with more complex classification models. As T_e increases, values of λ for which our model has equivalent accuracy to the edge classifier increases – Fig. 7. When T_e is higher, the expected cost of a request is lower under the same level of confidence in prediction. This leads to higher values of λ that still have a large communication saving over the edge baseline.

3) Key use-cases identified: The appropriateness of our method depends on the application context. We can conclude that our proposed strategy shows most potential where:

- Network topology contains constrained communication channels between distributed devices.
- The cost of edge server offloads are very high.
- The data have sufficient separation to allow for many instances of direct reports to the mobile device, and sufficient overlap that there are enough instances where data requests commonly lead to successful classification.

V. Conclusion & Future Research

We define a hybrid mechanism that makes greedy local decisions at end devices, to enable operation between different modes of inference, leading to a communication cost and accuracy trade-off. Experimental results demonstrate the efficacy of our proposed mechanism on a real-world IoT dataset. We expect this mechanism will have broader application in areas such as multi-agent perception in vehicular networks [9] where characteristics we discuss in Section IV-D3 could apply. Our research leaves several opportunities for future research:

With temporal and non-independent data we expect different results. More accurate estimates of communication cost using simulations of real-world network conditions may be beneficial.

We make local greedy decisions as to where to process or offload data. In principle, the data available at the edge device is inversely proportional to the number of local confident predictions – hence we expect good accuracy, as we consistently observe in our experiments. Accuracy *could* be severely affected in cases where many low-quality contradictory predictions from individual sensors deprive the edge device of sufficient data for quality predictions. Further investigation is needed to determine the circumstances and severity of when this occurs.

Finally, in order to minimize a system's overall power consumption we need to account for power consumption of inference and use better metrics of communication cost.

References

[1] E. Baccour, N. Mhaisen, A. A. Abdellatif, A. Erbad, A. Mohamed, M. Hamdi, and M. Guizani, "Pervasive AI for IoT Applications: A Survey on Resource-Efficient Distributed Artificial Intelligence," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2366–2418, 2022.

[2] T. Alsboui, Y. Qin, R. Hill, and H. Al-Aqrabi, "Distributed Intelligence in the Internet of Things: Challenges and Opportunities," *SN Computer Science*, vol. 2, p. 277, May 2021.

[3] I. A. Hashem, A. Siddiqa, F. A. Alaba, M. Bilal, and S. M. Alhashmi, "Distributed intelligence for IoT-based smart cities: a survey," *Neural Computing and Applications*, vol. 36, pp. 16621–16656, Sept. 2024.

[4] G. Kaur and R. S. Batth, "Edge Computing: Classification, Applications, and Challenges," in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 254–259, Apr. 2021.

[5] R. Dautov and S. Distefano, "Three-level hierarchical data fusion through the IoT, edge, and cloud computing," in *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, (Liverpool United Kingdom), pp. 1–5, ACM, Oct. 2017.

[6] J. Shao, Y. Mao, and J. Zhang, "Task-Oriented Communication for Multidevice Cooperative Edge Inference," *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 73–87, 2022.

[7] E. Gambi, G. Temperini, R. Galassi, L. Senigagliesi, and A. De Santis, "ADL Recognition Through Machine Learning Algorithms on IoT Air Quality Sensor Dataset," *IEEE Sensors Journal*, vol. 20, pp. 13562–13570, Nov. 2020.

[8] J. Li, H. Wei, J. Liu, and W. Liu, "FSLEdge: An energy-aware edge intelligence framework based on Federated Split Learning for Industrial Internet of Things," *Expert Systems with Applications*, vol. 255, p. 124564, Dec. 2024.

[9] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-Agent Perception via Communication Graph Grouping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4106–4115, 2020.

[10] Y. Huang, X. Qiao, S. Dustdar, J. Zhang, and J. Li, "Toward Decentralized and Collaborative Deep Learning Inference for Intelligent IoT Devices," *IEEE Network*, vol. 36, pp. 59–68, Jan. 2022.

[11] Z. Zhao, K. M. Barijough, and A. Gerstlauer, "DeepThings: Distributed Adaptive Deep Learning Inference on Resource-Constrained IoT Edge Clusters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, pp. 2348–2359, Nov. 2018.

[12] C. Luo, J. Chen, X. Feng, J. Zhang, and J. Li, "BSL: Sustainable Collaborative Inference in Intelligent Transportation Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, pp. 15995–16005, Dec. 2023.

[13] D. Simões, N. Lau, and L. P. Reis, "Multi Agent Deep Learning with Cooperative Communication," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 10, pp. 189–207, July 2020.

[14] C. Zhang and V. Lesser, "Coordinating Multi-Agent Reinforcement Learning with Limited Communication," *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 1101–1108, 2013.