IEEE ComSoc IEEE Communications Society | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received 14 August 2023; revised 22 December 2023; accepted 5 February 2024.
Date of publication 14 February 2024; date of current version 22 February 2024.

The associate editor coordinating the review of this article and approving it for publication was C. Shen.

Digital Object Identifier 10.1109/TMLCN.2024.3366501

Getting the Best Out of Both Worlds: Algorithms for Hierarchical Inference at the Edge

VISHNU NARAYANAN MOOTHEDATH[1], JAYA PRAKASH CHAMPATI[2] (Member, IEEE), AND JAMES GROSS[1] (Senior Member, IEEE)

[1]Department of Intelligent Systems, KTH Royal Institute of Technology, 11428 Stockholm, Sweden
[2]Edge Networks Group, IMDEA Networks Institute, Leganes, 28918 Madrid, Spain

CORRESPONDING AUTHOR: V. N. MOOTHEDATH (vnmo@kth.se)

This work was supported in part by the Verket för Innovationssystem (VINNOVA) Competence Center for Trustworthy Edge Computing Systems and Applications (TECoSA) at KTH Royal Institute of Technology, in part by the Vetenskap Radet (VR)–Optimal Sampling for Interactive Networked Applications under Grant 2022-03922, and in part by the European Union through Marie Skłodowska-Curie Actions - Postdoctoral Fellowships (MSCA-PF) Project "DIME: Distributed Inference for Energy-efficient Monitoring at the Network Edge" under Grant 101062011.

ABSTRACT We consider a resource-constrained Edge Device (ED), such as an IoT sensor or a microcontroller unit, embedded with a small-size ML model (S-ML) for a generic classification application and an Edge Server (ES) that hosts a large-size ML model (L-ML). Since the inference accuracy of S-ML is lower than that of the L-ML, offloading all the data samples to the ES results in high inference accuracy, but it defeats the purpose of embedding S-ML on the ED and deprives the benefits of reduced latency, bandwidth savings, and energy efficiency of doing local inference. In order to get the best out of both worlds, i.e., the benefits of doing inference on the ED and the benefits of doing inference on ES, we explore the idea of *Hierarchical Inference* (HI), wherein S-ML inference is only accepted when it is correct, otherwise the data sample is offloaded for L-ML inference. However, the ideal implementation of HI is infeasible as the correctness of the S-ML inference is not known to the ED. We thus propose an online meta-learning framework that the ED can use to predict the correctness of the S-ML inference. In particular, we propose to use the probability corresponding to the maximum probability class output by S-ML for a data sample and decide whether to offload it or not. The resulting online learning problem turns out to be a Prediction with Expert Advice (PEA) problem with continuous expert space. For a full feedback scenario, where the ED receives feedback on the correctness of the S-ML once it accepts the inference, we propose the HIL-F algorithm and prove a sublinear regret bound $\sqrt{n \ln(1/\lambda_{\min})/2}$ without any assumption on the smoothness of the loss function, where n is the number of data samples and $\lambda_{\min}$ is the minimum difference between any two distinct maximum probability values across the data samples. For a no-local feedback scenario, where the ED does not receive the ground truth for the classification, we propose the HIL-N algorithm and prove that it has $O\left(n^{2/3} \ln^{1/3}(1/\lambda_{\min})\right)$ regret bound. We evaluate and benchmark the performance of the proposed algorithms for image classification application using four datasets, namely, Imagenette and Imagewoof, MNIST, and CIFAR-10.

INDEX TERMS Hierarchical inference, edge computing, regret bound, continuous experts.

I. INTRODUCTION

EMERGING applications in smart homes, smart cities, intelligent manufacturing, autonomous internet of vehicles, etc., are increasingly using Deep Learning (DL) inference. Collecting data from the Edge Devices (EDs) and performing remote inference in the cloud results in bandwidth, energy, and latency costs as well as reliability (due to wireless transmissions) and privacy concerns. Therefore, performing local inference using embedded DL models, which we refer to as S-ML (Small-ML) models, on EDs has received significant research interest in the recent past [1], [2], [3]. These S-ML models range from DL models

that are optimised for moderately powerful EDs, such as mobile phones, to tinyML DL models that even fit on micro-controller units. However, S-ML inference accuracy reduces with the model size and can be potentially much smaller than the inference accuracy of large-size state-of-the-art DL models, which we refer to as L-ML (Large-ML) models, that can be deployed on Edge Servers (ESs). For example, for an image classification application, an S-ML can be a quantized *MobileNet* [4] with a width multiplier of 0.25, that has a memory size of 0.5 MB and an inference accuracy of 39.5% for classifying ImageNet dataset [5], whereas CoCa [6], an L-ML, has an accuracy of 91% and a memory size in the order of GBs.

One may choose to achieve the accuracy of L-ML model while utilising the computational capabilities of EDs using the well-known DNN partitioning techniques, e.g., see [7], [8], and [9]. Note that such partitioning techniques require processing time and energy profiling of the layers on EDs as well as on ESs to decide the optimal partition points. Early Exit is yet another technique that introduces side branches in between the layers of DL models to trade-off accuracy with latency [10]. In contrast to these techniques, in this work, we explore the novel idea of *Hierarchical Inference* (HI). Consider that an ED is embedded with an S-ML and an L-ML[1] is deployed on an ES (to which the ED enlists to get help for doing inference). In HI, we propose that an ED first observes the S-ML inference on each data sample and offloads it to L-ML only if S-ML inference is incorrect.

Clearly, the ambition of HI is to maximise the use of S-ML in order to reap the benefits of reduced latency, bandwidth savings, and energy efficiency while not losing inference accuracy by offloading strategically to L-ML, thus achieving the best benefits out of the two worlds: EDs and ESs. However, the central challenge is that the incorrect inferences are inherently unknown at the ED, and a decision under uncertainty needs to be taken. Thus, for each sample, we ask the question: should the ED accept the inference from S-ML or offload for inference from L-ML? In this work, we focus on the pervasive *classification applications* and address the above question as an online sequential decision problem by proposing a novel HI meta-learning framework, shown in Fig. 1. This framework facilitates the ED in deciding in real time whether an S-ML inference for a given sample should be accepted or rejected, where in the latter case the sample is offloaded for an accurate inference by the L-ML.

In our framework, for each sample, the HI learning algorithm observes p, the maximum probability value in the probability distribution over classes output by the S-ML. In a DNN for example, a softmax function such as sigmoid in the last layer outputs these probabilities.

It then decides to offload, receiving a fixed cost $0 \leq \beta < 1$, or not to offload, receiving a cost 0 if the inference is correct, and a cost 1, otherwise. We will show later that this

[1]Both S-ML and L-ML are trained ML models deployed for providing inference and HI does not modify these models.

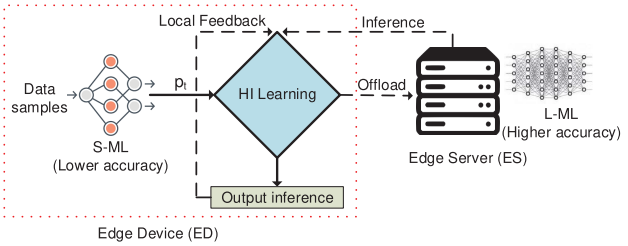


FIGURE 1. Schematic of the HI meta-learning framework.

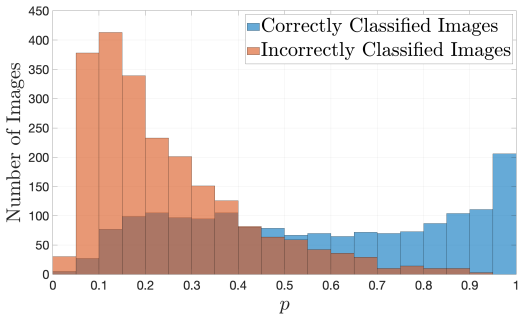


FIGURE 2. Classification of Imagenette by a small-size quantized MobileNet using width multiplier 0.25 [4].

cost structure facilitates HI by maximising the offloading of samples with incorrect inference and minimising offloading the samples with correct inference. We will also argue in Section III that any other arbitrary costs could be transformed into this particular cost structure. Here, the offload cost can be considered as the cost incurred for communication, latency, resource usage, etc., and the fluctuations in these from sample to sample could be captured using an average. We also assume that S-ML accepts the inference of L-ML as the ground truth implying that the top-1 accuracy of L-ML is 100%. We use this only for simplicity and is not necessary for our results. We will further discuss this assumption in Section III and also present the analysis without it later in Appendix E.

Intuitively, if the maximum probability p is high, then accepting S-ML inference will likely result in cost 0 and thus, it is beneficial to do so. However, if p is low, the cost will likely be equal to 1, and thus offloading with cost β is beneficial. This can be seen from Fig. 2, where we present the number of misclassified and correctly classified images of the dataset *Imagenette* [11] by the classifier *MobileNet* [4]. Observe that, for $p \geq 0.45$ (approximately) it is beneficial to accept the inference of MobileNet in the simple sense that there are more images correctly classified, and offloading is beneficial for $p < 0.45$. For specific costs, this can be easily done offline using a simple search if similar histograms are available, which is not the case in any real-time scenario. To provide an intuitive idea, one can visually find out this optimum threshold (for $\beta = 0.5$) by simply finding a point such that the total number of incorrectly classified images (i.e., the brown bars) below it is equal to the total number of images (i.e., the brown and blue bars) above it. The problem

TABLE 1. List of abbreviations.

ED	edge device	HI	hierarchical inference	HIL-F	HIL algorithm: full feedback
ES	edge server	HIL	hierarchical inference learning	HIL-N	HIL algorithm: no-local feedback
S-ML	small-size ML	PEA	prediction with expert advice	EWF	exponentially weighted forecaster
L-ML	large-size ML	MAB	multi-armed bandit	DNN	deep neural network

that we pose is finding the optimum threshold in an online manner.

The above problem falls in the domain of Prediction with Expert Advice (PEA) [12]. However, given the continuous expert space (or action space) for θ, as elaborated in Section IV, the standard Exponentially Weighted Average Forecaster (EWF) cannot be used here. Additionally, another challenge is that the local cost is unobservable when the S-ML inference is accepted due to the unavailability of the ground truth at ED. This situation characterizes a *no-local feedback* scenario. For simplify the solution, we initially consider a *full feedback* scenario assuming local feedback is available, and then adapt the solution to the more realistic no-local feedback scenario, resulting in algorithms designated as HIL-F and HIL-N for the full feedback and no-local feedback scenarios, respectively.

A novel aspect of our algorithms is that they use the structural properties of the HI learning problem at hand to find a set of non-uniform intervals obtained by doing dynamic and non-uniform discretisations and use these intervals as experts, thereby transforming the problem from a continuous to a discrete domain without introducing any error due to this discretisation. To the best of our knowledge, our work is the first attempt to extend the concept of continuous experts to the no-local feedback scenario and find regret bounds for the same. We summarise our main contributions below.

- We propose a novel meta-learning framework for HI that decides whether a data sample that arrived should be offloaded or not based on S-ML output. For the full feedback scenario, we prove that $O\left(\sqrt{n \log n}\right)$ is the lower bound for the regret bound that can be achieved by any randomised algorithm for a general loss function, where n is the number of data samples.
- We propose the HIL-F (HI Learning with Full Feedback) algorithm that uses exponential weighting and dynamic non-uniform discretisation. We prove that HIL-F has $\sqrt{n \ln(1/\lambda_{\min})/2}$ regret bound, where $\lambda_{\min}$ is the minimum difference between any two **distinct** p values among the n samples.
- We propose HIL-N (HI Learning with no-local feedback) algorithm, which on top of HIL-F, uses an unbiased estimate of the loss function. We prove a regret bound $O\left(n^{2/3} \ln^{1/3}(1/\lambda_{\min})\right)$. We discuss the ways to approximate $\lambda_{\min}$ and find the optimal values of the parameters used.
- We show that the computational complexity of our algorithms in round t is $O\left(\min(t, \frac{1}{\lambda_{\min}})\right)$.
- Finally, we evaluate the performance of the proposed algorithms for an image classification application using four datasets, namely, Imagenette and Imagewoof [11], MNIST [13], and CIFAR-10 [14], [15]. For the first two, we use MobileNet, for MNIST, we implement a linear classifier, and for CIFAR-10, we use a readily available CNN. We compare with four baselines – the optimal fixed-θ policy, one that offloads all samples, one that offloads none, and a hypothetical genie algorithm that knows the ground truth.

In our recent work [16], we have provided a more general definition of HI and provided multiple use cases, and also compared HI with existing DL inference approaches at the edge. However, in [16], we used a fixed threshold for offloading without the online learning aspects that we do in this work. Further, unlike [16], in this work we rely heavily on analytical results and discuss how close the solution is to an offline optimum.

This paper is organised as follows: In Section II we go through the related research and explain the novelty in the contributions. In Section III, we describe the system model followed by some background information and preliminary results in Section IV. Sections V and VI details HIL-F and HIL-N, derive their regret bounds, and Section VII discuss their computational complexity. Finally, we show the numerical results in Section VIII and conclude in Section IX.

We summarise some important abbreviations in TABLE 1.

II. RELATED WORK

A. INFERENCE OFFLOADING

Since the initial proposal of edge computing in [17], significant attention was given to the computational offloading problem, wherein the ED needs to decide which jobs to offload and how to offload them to an ES. The majority of works in this field studied offloading generic computation jobs, e.g., see [18], [19], and [20]. In contrast, due to the growing interest in edge intelligence systems, recent works studied offloading data samples for ML inference both from a theoretical [21], [22], [23] and practical [24], [25] perspectives. In [21], offloading between a mobile device and a cloud is considered. The authors account for the time-varying communication times by using model selection at the cloud and by allowing the duplication of processing the job at the mobile device. In [22], the authors considered a scalable-size ML model on the ED and studied the offloading decision to maximise the total inference accuracy subject to a time constraint. All the above works focus on dividing the load of the inference and do not consider HI and online learning. Our

work is in part motivated by [23], where the authors assumed that the energy consumption for local inference is less than the transmission energy of a sample and studied offloading decision based on a confidence metric computed from the probability distribution over the classes. However, in contrast to our work, the authors do not consider the meta-learning framework and compute a threshold for the confidence metric based on the energy constraint at the ED.

B. ON-DEVICE INFERENCE

Several research works focused on designing S-ML models to be embedded on EDs that range from mobile phones to microcontroller units. While optimisation techniques such as parameter pruning and sharing [26], weights quantisation [27], and low-rank factorisation [28] were used to design the S-ML models, techniques such as EarlyExit were used to reduce the latency of inference. For example, [29] studied the use of DNNs with early exits [10] on the edge device, while [30] studied the best DNN selection on the edge device for a given data sample to improve inference accuracy and reduce latency. These works do not consider inference offloading and in turn HI.

C. DNN PARTITIONING

Noting that mobile devices such as smartphones are embedded with increasingly powerful processors and the data transmitted between intermediate layers of a DNN is much smaller than the input data in several applications, the authors in [7] studied partitioning DNN between a mobile device and cloud to reduce the mobile energy consumption and latency. Following this idea, significant research work has been done that includes DNN partitioning for more general DNN structures under different network settings [8], [31] and using heterogeneous EDs [9], among others. In contrast to DNN partitioning, under HI, ED and ES may import S-ML and L-ML algorithms from the pool of trained ML algorithms available on open-source libraries such as Keras, TFLite, and PyTorch. Furthermore, HI doesn't even require that S-ML and L-ML be DL models but rather can even be signal processing algorithms. On the one hand, there is significant research by the tinyML community for building small-size DNNs that can be embedded on micro-controllers and also in designing efficient embedded hardware accelerators [2]. On the other hand, abundant state-of-the-art DNNs are available at edge servers that provide high inference accuracy. Our work is timely as HI will equip ML at the edge to reap the benefits of the above two research thrusts. To the best of our knowledge, we are the first to propose an online meta-learning framework for HI.

D. ONLINE LEARNING

The problem of minimising the regret, when the decision is chosen from a finite expert space falls under the well-known Prediction with Expert Advice (PEA) or Multi-Armed Bandit (MAB) problems [12], [32]. We will explain more about these problems in Section IV. We will see that we cannot directly use these problems due to the uncountable nature of the expert space in our problems which we will elaborate in III. We will also explain why some of the existing literature on continuous extensions of PEA and MAB are not suited or sub-optimal for our specific problem.

E. CLASSIFICATION WITH REJECTION

In the machine learning community, classification with rejection methods, which accept only the confident inferences and reject (equivalent to 'offload' in our problem) the rest, has been well studied in the literature. In the survey paper [33], the authors elaborate on these methods in detail and discuss several confidence metrics. One could potentially use an S-ML model with a rejection option to directly facilitate the offloading decision. However, S-ML models with a reject option come at the expense of higher resource requirements. For example, training a multi-class classifier with m classes with a reject option requires training m binary classifiers [34]. For applications like image classification which have a large number of classes (e.g. ImageNet has 1000 classes) loading such a model with the rejection option will be prohibitive for resource-constrained devices. In contrast, we use the basic confidence metric p for offloading or accepting decisions. Our framework is quite flexible in that, any off-the-shelf ML models can be used as the S-ML. Further, unlike classification with rejection mode, our online learning approach for p will potentially benefit the decisions even when the samples are generated out-of-distribution.

In [35], it was shown that the probability of the maximum probability class is a very strong confidence metric for the detection of potential errors. As mentioned already, we use this metric p throughout the paper. However, one could also use other metrics such as the difference between the first and the second largest probabilities [36], [37]. In [33], the authors also discuss calibrated models where the maximum probability also reflects the actual likelihood of correctness of the corresponding class. Note that in our HI framework, S-ML need not be calibrated, and the proposed algorithm and analysis apply to any confidence metric.

III. SYSTEM MODEL AND PROBLEM STATEMENT

We consider the system shown in Fig. 1, with an ED enlisting the service of an ES for data classification applications. For the EDs, we focus on resource-constrained devices such as IoT sensors or microcontroller units. The ED is embedded with an S-ML which provides lower *inference accuracy*, i.e., the top-1 accuracy, whereas the ES runs an L-ML with higher accuracy. For example, for an image classification application, an S-ML can be a quantized MobileNet [4] with a width multiplier of 0.25; its memory size is 0.5 MB and has an inference accuracy of 39.5% for classifying ImageNet dataset [5], whereas CoCa [6], an L-ML, has accuracy 91% and has memory size in the order of GBs. The only assumption that we make on the algorithms is that the L-ML is significantly more accurate and costlier than the S-ML. In this paper,

the S-ML or the L-ML algorithms can be any classification algorithm including regression algorithms, SVMs, random forests, and DNNs.

Given an arbitrary sequence of n data samples ariving over time at the ED. We assume that each sample first goes through local inference and the decision is made according to the inference results and parameters. This is an essential assumption to facilitate HI, otherwise, the ED cannot infer anything about the sample. As stated earlier in Section I, we assume that all the offloaded images will be correctly classified by the L-ML. This assumption is not necessary, and we provide an extended analysis in Appendix E without this assumption where we consider an imperfect L-ML and include a cost of incorrect inferences at L-ML. The assumption, however, is justified in practice because the ED doesn't have the ground truth and there is no meaningful method for ED (or ES) to check whether the L-ML inference is correct or not. Therefore, it is reasonable that it aims to achieve an inference accuracy as close as possible to that of L-ML by treating the output of the L-ML as the ground truth.

A. SYSTEM COSTS AND FEEDBACK SETTINGS

Let t denote the index of a data sample (e.g., an image), or simply a sample, that arrives t^{th} in the sequence. Let p_t denote the maximum probability in the probability distribution over the classes output by S-ML for the sample t and the class corresponding to p_t is declared as the true class for computing the top-1 accuracy, which is very typical in a wide variety of classifier algorithms.[2] Let binary random variable Y_t denote the cost representing the ground truth that is equal to 0 if the class corresponding to p_t is the correct class and is equal to 1, otherwise. Clearly, given an S-ML model, Y_t depends on p_t and the sample.

Let $\beta \in [0, 1)$ denote the cost incurred for offloading the image for inference at the ES. This cost, for example, may include the costs for the transmission energy and the idle energy spent by the transceiver till the reception of the inference. Note that, if $\beta \geq 1$, then accepting the inference of S-ML, which incurs a cost at most 1, for all samples will minimise the total cost. This particular cost structure of $\{0, \beta, 1\}$ is chosen for easy computations. However, note that any other set of arbitrary costs can be transformed into this form by ignoring the common and hence non-optimisable costs and properly scaling the rest. To understand this, assume the cost of correct S-ML Inference, the cost of offload, and the cost of incorrect inference are C_0, C_β, and C_1, respectively. Also assume that $C_0 < C_\beta < C_1$, and that the cost of S-ML inference C_0 is a common component that is incurred irrespective of the decision and inference outcome. Thus, subtracting this common component and then dividing by $C_1 - C_0$ gives us a zero cost for correct local inference, a unit cost for local incorrect inference, and a cost of $\beta \in (0, 1)$ for offload given by

$$\beta = \frac{C_\beta - C_0}{C_1 - C_0}. \tag{1}$$

Throughout this paper, we consider the offload cost β to be a constant known apriori. However, it turns out that a varying beta, β_t, $t = 1, 2, \ldots,$ could also provide similar results, with β replaced with the expectation of the sequence $\{\beta_t\}$. This is elaborated as a remark at the end of Appendix E.

As explained in Section I, in round t, we use the following decision rule $\mathfrak{D}_t$ based on the choice of threshold $\theta_t \in [0, 1]$:

$$\mathfrak{D}_t = \begin{cases} \text{Do not offload} & \text{if } p_t \geq \theta_t, \\ \text{Offload} & \text{if } p_t < \theta_t. \end{cases} \tag{2}$$

Given p_t, choosing threshold θ_t thus results in a cost/loss $l(\theta_t, Y_t)$ at step t, where we omit the variable p_t from the function for simplicity in notations. This is given by

$$l(\theta_t, Y_t) = \begin{cases} Y_t & p_t \geq \theta_t, \\ \beta & p_t < \theta_t. \end{cases} \tag{3}$$

We use boldface notations to denote vectors. Let $\boldsymbol{Y}_t = \{Y_\tau\}, \boldsymbol{\theta}_t = \{\theta_\tau\}$, and $\mathbf{p}_t = \{p_\tau\}$, $\tau = 1, 2, \ldots, t \leq n$. Further, let $\boldsymbol{Y} := \boldsymbol{Y}_n$, $\boldsymbol{\theta} := \boldsymbol{\theta}_n$ and $\mathbf{p} := \mathbf{p}_n$ for convenience. Finally, we define $\lambda_{\min}$ as the minimum difference between any two distinct probability values in the sequence $\boldsymbol{p}_n$. Define the cumulative cost $L(\boldsymbol{\theta}, \boldsymbol{Y})$ as $L(\boldsymbol{\theta}, \boldsymbol{Y}) = \sum_{t=1}^{n} l(\theta_t, Y_t)$.

1) FEEDBACK SETTINGS

Recall that the ground truth, available at the ES by virtue of the perfect L-ML, is fed back to the ED for all offloaded samples. However, at the ED, the ground truth (Y_t) is not accessible, requiring further modifications and/or assumptions to learn the accuracy of S-ML. In this paper, we consider two scenarios: *full feedback* and *no-local feedback*. The no-local feedback scenario, realistic without additional assumptions, utilises the exploration-exploitation tradeoff to acquire ground truth, by offloading a subset of the samples where a decision to accept the S-ML inference is made.

Primarily serving as an analytical precursor to the subsequent discussion, we also present the full feedback scenario, assuming the ground truth's availability at the Edge Device (ED) without any exploration. We start the analysis with this scenario, helping the reader in comprehending the solution approach and subsequently extending the understanding to the more realistic no-local feedback scenario with relative ease. It is worth noting that one may still imagine scenarios that closely resembles a full feedback scenario. Hypothetical examples include the system acting as an assistant to a human user in classification, allowing for a binary inference on classification correctness, or a delayed provision of ground truth due to latency or privacy concerns.

B. PROBLEM STATEMENT

We are interested in devising online algorithms for learning the optimal threshold that strikes a balance between reducing

[2]Our framework allows other confidence metrics besides p_t and it does not involve any further modification in the remainder of this work.

TABLE 2. List of symbols.

p_t	S-ML maximum probability output	β	normalised offload cost	θ_t	decision threshold
Y_t	random variable denoting inference correctness	$l(\cdot,\cdot)$	loss function	$L(\cdot,\cdot)$	cumulative loss function
W_t	weight function normalisation factor	$w_t(.)$	weight function	R_n	regret
$\lambda_{\min}$	minimum difference between two p_t values	q_t	probability of offloading	$p_{[i]}$	i^{th} smallest disctinct p_t
$Z_t(\epsilon)$	Bernaulli exploration variable with parameter ϵ	η	learning rate	$\tilde{l}(\cdot,\cdot)$	pseudo loss function

the number of offloaded images and improving inference accuracy, thereby enhancing the responsiveness and energy efficiency of the system. Let $\boldsymbol{\theta^*} = \{\theta^*, \theta^*, \dots\}$, a vector of size n with all values θ^*, denote an optimal fixed-θ policy and $L(\boldsymbol{\theta}^*, \boldsymbol{Y})$ denote the corresponding cost. Then,

$$L(\boldsymbol{\theta}^*, \boldsymbol{Y}) = \sum_{t=1}^{n} l(\theta^*, Y_t),$$

where θ^* need not necessarily be unique and is given by

$$\theta^* = \underset{\theta\in[0,1]}{\operatorname{argmin}} \sum_{t=1}^{n} l(\theta, Y_t).$$

Given a sequence $\boldsymbol{Y}$, define the regret as

$$R_n = \mathbb{E}_\pi\left[L(\boldsymbol{\theta}, \boldsymbol{Y})\right] - L(\boldsymbol{\theta}^*, \boldsymbol{Y}), \tag{4}$$

where the expectation $\mathbb{E}_\pi[\cdot]$ is with respect to the distribution induced by an arbitrary algorithm π.

We aim to develop HIL-F (HI Learning with full feedback) and HIL-N (HI Learning with no-local feedback) algorithms for the two scenarios, each with a sublinear upper bound (i.e., a bound approaching 0 as n goes to ∞) on $\mathbb{E}_{\boldsymbol{Y}}[R_n]$ – the expected regret over the distribution of all possible sequences $\boldsymbol{Y}$. We refer to this bound as an expected regret bound. Note that a regret bound applicable to any given sequence $\boldsymbol{Y}$ extends to the expected regret (or even the maximum regret) over all possible sequences of $\boldsymbol{Y}$. Consequently, for simplicity, we restrict the analysis to a given $\boldsymbol{Y}$ in the upcoming sections. However, in the numerical section, we will present results with the expected average regret $\mathbb{E}_{\boldsymbol{Y}}[\frac{1}{n}R_n] = \frac{1}{n}\mathbb{E}_{\boldsymbol{Y}}[R_n]$ and the expected average cost $\frac{1}{n}\mathbb{E}_{\boldsymbol{Y},\pi}[L(\boldsymbol{\theta}, \boldsymbol{Y})]$. The averaging over the number of samples n normalizes the maximum to 1, facilitating easy comparison and removing the dependency on the size of different datasets.

We summarise all the relevant notations in TABLE 2.

IV. BACKGROUND AND PRELIMINARY ANALYSIS

A. LEARNING PROBLEMS

The HI learning problem falls into the category of PEA [12] problems. In the standard PEA problem, N experts (or actions) are available for a predictor – known formally as a *forecaster*. When the forecaster chooses an expert, it receives a cost/reward corresponding to that expert. If the cost is only revealed for the chosen expert, then this setting is the MAB. In contrast to the standard PEA, we have an uncountable expert space where the expert θ_t belongs to the continuous space [0, 1]. Continuous action space is well studied in MAB settings, e.g., see [38], [39], and [40], where the main technique used is to discretise the action space and bound the regret by assuming that the unknown loss function has smoothness properties such as uniformly locally Lipschitz. However, the problem at hand does not assume any smoothness properties for the loss function.

As discussed briefly in Section I, one well-known forecaster for standard PEA is the exponential weighted average forecaster (EWF). For each expert, EWF assigns a weight that is based on the cost incurred for choosing this expert. During each prediction, EWF selects an expert with a probability computed based on these weights. It is known that for n predictions, EWF achieves a regret $\sqrt{n \ln N/2}$. However, the continuous nature of the expert space renders EWF not directly usable for solving the problem at hand. An extension of EWF was considered in [41], and a regret bound for convex losses is obtained for continuous experts, conditioned on a hyperparameter $\gamma > 0$. Later, a particular γ is proposed to get the optimum regret bound of $1 + \sqrt{n \ln n/2}$. We, on the other hand, do not require any hyperparameter and, more importantly, do not assume any convexity for the loss function. In addition, [41] does not describe how to compute the integral required for computing the weights. Furthermore, the solution in [41] is only applicable to HIL-F with full feedback, but not to HIL-N in which case ours is the first work to the best of our knowledge.

One may discretise [0, 1] with a uniform interval length Δ and use the standard EWF, where a straightforward substitution of the number of experts $N = 1/\Delta$ results in a regret bound of $\sqrt{n \ln(1/\Delta)/2}$. However, to not sacrifice the accuracy due to this discretisation, one has to take Δ small enough such that no two probability realisation p_t falls within an interval. This is to make sure that the cumulative loss function is constant within each interval, which will be more clear after Lemma 1. Thus, if $\lambda_{\min}$ is the minimum separation between any two distinct probabilities p_t, $1 \le t \le n$, the best attainable regret bound of a standard EWF using uniform discretisation is $\sqrt{n \ln(1/\lambda_{\min})/2}$ with $N = 1/\lambda_{\min}$. We will soon see that these regret bounds are similar to what we get using our proposed algorithms, but the added complexity with a large number of experts from the first round onwards makes it sub-optimal.

In this paper, we start with the continuous experts and then use the structure of the problem to formulate it in a discrete domain. We propose a non-uniform discretisation that retains the accuracy of a continuous expert while reducing the complexity to the theoretical minimum with at most

$n+1$ experts after n^{th} round. Note that, due to the non-uniform discretisation, the proposed HIL does not involve Δ, but instead involves $\lambda_{\min}$, where $1/\lambda_{\min}$ acts similar to N in the regret bound. In Section V, we provide simple methods to approximate $\lambda_{\min}$.

B. PRELIMINARY ANALYSIS

To choose a good threshold θ_t in round t, we take a hint from the discrete PEA [12] where a weight for an expert is computed using the exponential of the scaled cumulative losses incurred. We extend this idea and define continuous weight function $w_t(\theta)$ as follows:

$$w_{t+1}(\theta) = e^{-\eta \sum_{\tau=1}^{t} l(\theta, Y_\tau)}$$
$$= e^{-\eta \sum_{\tau=1}^{t-1} l(\theta, Y_\tau)} e^{-\eta l(\boldsymbol{\theta}, Y_t)} \quad (5)$$
$$= w_t(\theta) e^{-\eta l(\theta, Y_t)}. \quad (6)$$
$$W_{t+1} = \int_0^1 w_{t+1}(\theta)\, \mathrm{d}\theta. \quad (7)$$

Here, $\eta > 0$ is the learning rate and W is the normalisation factor. At each round t, the normalised weights give the probability distribution for choosing the next threshold θ_{t+1}, and thus they can be used to learn the system. However, it comes with two challenges – (i) finding a (set of) thresholds that follow this distribution and (ii) computing the integral. Although these challenges can be solved using direct numerical methods, they incur a large amount of computational cost. For instance, the inverse transformation method can generate a random sample of the threshold with this distribution. Instead, we use the facts from (2) and (3) that our final decision (to offload or not) depends solely on the relative position of θ_t and p_t, but not directly on θ_t. Thus, using the distribution given by the normalised weights, we define q_t as the probability of *not* offloading, i.e., the probability that θ_t is less than p_t, where

$$q_t = \frac{\int_0^{p_t} w_t(x)\, \mathrm{d}x}{W_t}. \quad (8)$$

Thus, the decision $\mathfrak{D}_t$ from (2) boils down to *do not offload* and *offload* with probabilities q_t and $(1-q_t)$, respectively.

Having addressed the first challenge, our focus shifts to finding efficient methods for computing the integral in (8). It's worth noting that the cumulative loss function, $L(\boldsymbol{\theta}_t, \boldsymbol{Y}_t) = \sum_{\tau=1}^{t} l(\theta_\tau, Y_\tau)$, can potentially take 3^t different values (due to 0, 1, or β cost in each step) without a necessary pattern, making direct analytical integration impractical. To overcome this challenge, we utilize Lemma 1 and transform the integral into a summation by discretizing the domain [0, 1] into a finite set of non-uniform intervals.

The non-uniform discretisation suggested by the lemma is incremental and a new interval is (potentially) added in each round. Let's look at the structure of the weight function after n rounds. Let $p_0 = 0$ and $p_N = 1$, where N is the number of intervals formed in [0, 1] by the sequence of probabilities $\boldsymbol{p}_n$. Here, we have $N \leq n+1$ because of the repeated probabilities that do not result in the addition of a new interval. We denote these intervals by $B_i = (p_{[i-1]}, p_{[i]}], 1 \leq i \leq N$, where $p_{[i]}$ denotes the i^{th} smallest distinct probability in $\boldsymbol{p}_n$. Let $m_i, 1 \leq i \leq N$ be the number of times $p_{[i]}$ is repeated in $\boldsymbol{p}_n$. For instance, $N = n+1$ and $p_{[i]} = p_i$ iff $m_i = 1 \forall i$. Finally, let $Y_{[i]}, i = 1, 2, \ldots n$ be the i^{th} element in the ordered set of local inference costs ordered according to the increasing values of the corresponding probability p_i. Note that, i in $Y_{[i]}$ goes up to n while i in $p_{[i]}$ goes only up to N because any two local inference costs Y_j and Y_k associated with repeated probability values $p_j = p_k$ are two different but i.i.d random variables.

Lemma 1: The function $L(\boldsymbol{\theta}, \boldsymbol{Y})$ is a piece-wise constant function with a constant value in each interval B_i. Furthermore, if there are no repetitions in the sequence $\boldsymbol{p}_n$, then

$$L(\boldsymbol{\theta}^*, \boldsymbol{Y}) = \min_{1 \leq i \leq n+1} \left\{ (i-1)\beta + \sum_{k=i}^{n} Y_{[k]} \right\}.$$

Proof: By definition, p_t falls on the boundary of B_i, $\forall t$, for some i. Hence, B_i is a subset of either $(0, p_t]$ or $(p_t, 1]$.

$$\Rightarrow l(\theta_t, Y_t) = \begin{cases} Y_t, & \forall \theta : \theta_t \in B_i \subset (0, p_t], \text{ and} \\ \beta, & \forall \theta : \theta_t \in B_i \subset (p_t, 1]. \end{cases} \quad (9)$$

Thus, $\forall\, i \leq N$, $l(\theta, Y_t) := l(B_i, Y_t), \forall \theta \in B_i$. That is, the cost for all θ within an interval B_i takes a constant value of $l(B_i, Y_t)$, and this value depends on whether $p_{[i]}$ (the upper boundary of B_i) is greater than p_t or not. To prove the second part, note that $L(\theta, \boldsymbol{Y}) = \sum_{t=1}^{n} l(B_i, Y_t) : \theta \in B_i$.

$$\Rightarrow L(\boldsymbol{\theta}^*, \boldsymbol{Y}) = \min_{\theta \in [0,1]} L(\theta, \boldsymbol{Y}) = \min_{1 \leq i \leq N} \sum_{t=1}^{n} l(B_i, Y_t).$$

$$\sum_{t=1}^{n} l(B_i, Y_t) = \sum_{t=1}^{n} [\beta\, \mathbb{1}(p_t < p_{[i]}) + Y_t\, \mathbb{1}(p_t \geq p_{[i]})]$$
$$= \beta \sum_{j=1}^{i-1} m_j + \sum_{k=1+\sum_{j=1}^{i-1} m_j}^{n} Y_{[k]}$$

$$\Rightarrow L(\boldsymbol{\theta}^*, \boldsymbol{Y}) = \min_{1 \leq i \leq N} \left\{ \beta \sum_{j=1}^{i-1} m_j + \sum_{k=1+\sum_{j=1}^{i-1} m_j}^{n} Y_{[k]} \right\} \quad (10)$$

When there are no repetitions, we can substitute $m_j = 1, \forall j$ in the above expression to complete the proof. ■

From Lemma 1, we infer that the weight function is constant within the intervals defined by $\boldsymbol{p}_t$ for any t, and we can compute the integral in (8) by adding multiple rectangular areas formed by the length of each interval B_i and the corresponding constant weight within it. Thus, by converting the integral of $w_t(\theta)$ in a continuous domain to a summation of areas of rectangles with non-uniform bases, we not only reduce the complexity but also do that without sacrificing the accuracy of the results. We will discuss more on the computational complexity later in Section VII. It is worth noting that the property of the piece-wise nature – given by the first part of Lemma 1 – is not only valid for the particular loss function $l(\theta_t, Y_t)$, but also for any other loss function with

a single decision boundary (as in (3)) and discrete costs on either side of this boundary. This becomes important when we use a modified loss function for finding the optimum decision boundary θ^* for the HIL-N case in Section VI.

Consider the scenario where $\boldsymbol{p}_n$ is known a priori. We can then use the standard EWF with N intervals with the cost corresponding to interval B_i as defined in (9). The following Corollary states the regret bound for this algorithm.

Corollary 1: If the sequence $\boldsymbol{p}_n$ is known a priori, an EWF that uses the intervals B_i as experts achieves $\sqrt{n \ln N/2}$ regret bound. Consequently, given that $N = O(n)$, the regret bound of EWF is $O(\sqrt{n \ln n})$.

Note that, for the standard PEA with N experts, $\sqrt{n \ln N/2}$ is the lower bound for the regret bound for any randomised algorithm [42]. Thus, Corollary 1 implies that for the problem at hand, under a general loss function, no randomised algorithm has regret bound lower than $O(\sqrt{n \ln n})$. Clearly, the lower bound $O(\sqrt{n \ln n})$ is much higher than the lower bound of PEA, where the number of experts is independent of n. This establishes the hardness of our problem, which is imparted due to the dynamic and increasing nature of the number of experts. Adding to the difficulty, $O(\sqrt{n \ln n})$ can only be achieved if $\boldsymbol{p}_n$ is known a priori, which is not the case in practice.

With all preliminaries covered, we now present the HIL algorithms and their regret bounds for full feedback and no-local feedback scenarios in Sections V and VI, respectively.

V. FULL FEEDBACK

In this section, we consider the full-feedback scenario, where the algorithm receives the ground truth Y_t for all the samples, including those that are not offloaded by accepting the S-ML inference. For this scenario, we present the HIL-F algorithm in Algorithm 1. Some algorithmic rules for the parameter updates are given later in Section VII. As explained in the previous section, given p_t, we compute q_t, the probability of not offloading. Once the decision is made using q_t, the costs are received and the weights are updated using (6) and (7). For simplicity, we denote the expected cost received by HIL-F in round t by $\bar{l}(Y_t)$ and is given by

$$\bar{l}(Y_t) = \mathbb{E}_{Q_t}[l(\theta_t, Y_t)] = Y_t q_t + \beta(1 - q_t),$$

where the expectation is with respect to the probability distribution dictated by q_t. Also, let $\bar{L}(\boldsymbol{Y}) = \sum_{t=1}^{n} \bar{l}(Y_t)$ denote the total expected cost after n rounds. In the theorem below, we provide a regret bound for HIL-F.

Theorem 1: For $\eta > 0$, HIL-F achieves the following regret bound:

$$R_n = \bar{L}(\boldsymbol{Y}) - L(\boldsymbol{\theta}^*, \boldsymbol{Y}) \leq \frac{1}{\eta} \ln \frac{1}{\lambda_{\min}} + \frac{n\eta}{8}. \tag{11}$$

Proof: Proof of Theorem 1 is given in Appendix A. ■

Here, η is the learning rate of the algorithm. To find η^*, the η that minimises the above regret bound, we differentiate the regret R_n in (11) to obtain

$$\eta^* = \sqrt{\frac{8 \ln(1/\lambda_{\min})}{n}}. \tag{12}$$

Substituting (12) in (11), we get $R_n = \sqrt{n \ln(1/\lambda_{\min})/2}$.

What remains is to find an approximation for $\lambda_{\min}$, which is possible through various methods. For instance, one can use the precision of the probability outputs, i.e., if the probability outputs are truncated to 6 decimal places, then we know that $\lambda_{\min} \geq 10^{-6}$. Further, some datasets and/or S-ML models come with specific $\lambda_{\min}$. For example, the probability output by MobileNet on the Imagenette dataset is 8-bit and hence the probabilities are integer multiples of $1/256$. Even in cases where all these methods fail, we see that a decent approximation for $\lambda_{\min}$ is $\hat{\lambda}_{\min} = 1/(n+1)$.

Algorithm 1 The HIL-F Algorithm for Full Feedback

```
Initialise: Set w_1(θ) = 1, ∀θ ∈ [0, 1] and N = 1.
for every sample in round t = 1, 2, ... do
    S-ML outputs p_t.
    Compute q_t using (7) and (8), and generate Bernoulli
    random variable Q_t with ℙ(Q_t = 1) = q_t.
    if Q_t = 1 then
        Accept the S-ML inference and receive cost Y_t.
    else
        Offload the sample and receive cost β.
    end if
    Find the loss function using (3).
    if p_t is not a repetition then
        Update the intervals by splitting the interval containing p_t, at p_t. Increment N by 1.
    end if
    Update the weights for all intervals using (6), based on the interval positions with respect to p_t.
end for
```

VI. NO-LOCAL FEEDBACK

Under no-local feedback, the cost is unknown once the inference of the S-ML is accepted. For this scenario, we use the randomisation idea used for label efficient prediction problem [43], which is a variant of the PEA, where the costs in each round are not revealed, unless they are inquired for, and there can only be at most m inquires that can be made. For this variant, EWF is modified as follows: in each round, a Bernoulli random variable Z is generated with probability ϵ. If $Z = 1$, then feedback is requested and the costs are revealed. However, for our problem, the algorithm for the label-efficient prediction problem is not applicable due to the aspect of continuous expert space. Further, we do not have the notion of inquiring about the costs at the ED. Instead, when $Z = 1$, the sample has to be offloaded to the ES with cost β irrespective of the original decision made using q_t. These samples provide the ED with the inference using which the ED computes the cost Y_t.

Algorithm 2 The HIL-N Algorithm

1: Initialise: Set $w_1(\theta) = 1, \forall \theta \in [0, 1]$ and $N = 1$.
2: **for** $t = 1, 2, \ldots$ **do**
3: S-ML outputs p_t.
4: Compute q_t using weights from (14) and (15) and generate Bernoulli random variables Q_t and Z_t with $\mathbb{P}(Q_t = 1) = q_t$ and $\mathbb{P}(Z_t = 1) = \epsilon$.
5: **if** $Q_t = 1$ **and** $Z_t = 0$ **then**
6: Accept the S-ML inference and receive cost Y_t (unknown).
7: **else**
8: Offload the sample and receive cost β.
9: **end if**
10: Find the pseudo loss function using (13).
11: **if** p_t is not a repetition **then**
12: Update the intervals by splitting the interval containing p_t at p_t. Increment N by 1.
13: **end if**
14: Update the weights for all intervals using (14), based on the interval positions with respect to p_t.
15: **end for**

To address the above aspects we follow the design principles of HIL-F and use non-uniform discretisation of the continuous domain and propose the HI algorithm for no-local feedback (HIL-N), which is presented in Algorithm 2. Even though HIL-N and HIL-F have a similar structure, the design of HIL-N is significantly more involved and has the following key differences with HIL-F. Firstly, in line 5 of Algorithm 2, a Bernoulli random variable Z_t is generated with probability ϵ. If $Z_t = 1$, then the sample is offloaded even if $Q_t = 1$, and thus Y_t is realised in this case. This step is used to control the frequency of additional offloads carried out to learn the ground truth Y_t. Secondly, instead of the loss function, the weights are updated using a *pseudo loss function* $\tilde{l}(\theta_t, Y_t)$ defined as follows:

$$\tilde{l}(\theta_t, Y_t) = \begin{cases} 0 & p_t \geq \theta_t, Z_t = 0; \ [\textit{Do Not Offload}] \\ \dfrac{Y_t}{\epsilon} & p_t \geq \theta_t, Z_t = 1; \ [\textit{Offload}] \\ \beta & p_t < \theta_t. \qquad [\textit{Offload}] \end{cases} \tag{13}$$

We also update the equations (6), (7) and (8) as follows:

$$w_{t+1}(\theta) = w_t(\theta) e^{-\eta \tilde{l}_t(\theta, Y_t)}, \tag{14}$$

$$W_{t+1} = \int_0^1 w_{t+1}(\theta)\, \mathrm{d}\theta, \text{ and} \tag{15}$$

$$q_t = \frac{\int_0^{p_t} w_t(x)\, \mathrm{d}x}{W_t}. \tag{16}$$

We emphasise that the pseudo loss function $\tilde{l}(\theta_t, Y_t)$ is used only as part of the HIL-N algorithm, and is not the actual cost incurred by the ED. The actual cost remains unchanged and it depends only on the offloading decision and the correctness of the inference if not offloaded. However, this actual incurred cost or the corresponding loss function $l(\theta_t, Y_t)$ is unknown for the no-local feedback scenario, whenever the sample is not offloaded and the local inference is accepted. This is precisely the reason to introduce the pseudo loss function $\tilde{l}(\theta_t, Y_t)$ which is known in each t, and can be used in the HIL-N algorithm to update the weights. Recall from Section V that in HIL-F, the cost incurred and the cost used to update the weights are the same, and the incurred cost is β if and only if $p_t < \theta_t$. However in HIL-N, we use the pseudo cost to update the weights, and thus the actual cost incurred can be equal to β even if $p_t \geq \theta_t$. However, we designed the pseudo-loss function such that

$$\mathbb{E}_Z \left[\tilde{l}(\theta_t, Y_t) \right] = l(\theta_t, Y_t). \tag{17}$$

Therefore, the pseudo loss function is an unbiased estimate of the actual loss function, a fact that we will facilitate our analysis. Further, with the addition of a random variable Q, the regret for HIL-N can be rewritten as

$$R_n = \mathbb{E}_{\boldsymbol{QZ}}[L(\boldsymbol{\theta}, \boldsymbol{Y})] - L(\boldsymbol{\theta}^*, \boldsymbol{Y}), \tag{18}$$

where $\mathbb{E}_{\boldsymbol{QZ}}[\cdot]$ is expectation with respect to random variables $\{Q_1, Q_2, \ldots, Q_n\}$ and Bernoulli random variable Z.

Theorem 2: For $\eta, \epsilon > 0$, HIL-N achieves the regret bound

$$R_n \leq n\beta\epsilon + \frac{n\eta}{2\epsilon} + \frac{1}{\eta} \ln(1/\lambda_{\min}). \tag{19}$$

Proof: Proof of Theorem 2 is given in Appendix B. ■

The bound in Theorem 2 neatly captures the effect of ϵ on the regret. Note that the term $n\beta\epsilon$ is a direct consequence of offloading sample t, when $Z_t = 1$. Additionally, it is noteworthy that the bound for HIL-N in Theorem 2 exhibits similarity to the previously obtained bound for HIL-F in Theorem 1. Both bounds share a comparable relationship with dependent parameters such as n, η, $\lambda_{\min}$, among others. The additional terms in the HIL-N bound, rendering it a looser bound, result from the exploration aspect, where the correctness of the inference is available only for a subset of rounds determined by the Bernoulli parameter ϵ.

We now minimise this bound for HIL-N and find the parameters that render the bound to be sublinear in n. Denote the bound in Theorem 2 by $g(\epsilon, \eta)$. We have,

$$g(\epsilon, \eta) = n\beta\epsilon + \frac{n\eta}{2\epsilon} + \frac{1}{\eta} \ln(1/\lambda_{\min}). \tag{20}$$

Lemma 2: The function $g(\epsilon, \eta)$ defined in (20) has a global minimum at (ϵ^*, η^*), where $\eta^* = \left(\frac{2\ln^2(1/\lambda_{\min})}{\beta n^2}\right)^{1/3}$ and $\epsilon^* = \sqrt{\frac{\eta}{2\beta}}$. At this minimum, we have,

$$g(\epsilon^*, \eta^*) = 3n^{2/3} \left(\frac{\beta \ln(1/\lambda_{\min})}{2}\right)^{1/3}.$$

Proof: Proof of Lemma 2 is given in Appendix C. ■

Substituting the optimum parameters given by the above Lemma in (20), we obtain a sublinear regret bound for HIL-N. This is given in the following corollary.

Corollary 2: With $\eta = \left(\frac{2\ln^2(1/\lambda_{\min})}{\beta n^2}\right)^{1/3}$ and $\epsilon = \min\{1, \sqrt{\frac{\eta}{2\beta}}\}$, HIL-N achieves a regret bound sublinear in n:

$$R_n \leq 3n^{2/3}\left(\frac{\beta \ln(1/\lambda_{\min})}{2}\right)^{1/3}$$

Proof: Proof of Corollary 2 is given in Appendix D. ■

Remarks: It is worth noting the following:

1) The proof steps in Theorem 1 closely follow some analysis of the standard EWF for PEA with the added complexity to account for the continuous experts and non-uniform discretisation. The analysis for HIL-N is novel. In particular, the design of the unbiased estimator, steps 1 and 3 in the proof of Theorem 2, and the proof of Lemma 2 have new analysis.
2) The computational complexity of HIL-N is of the same order as that of HIL-F due to the similar interval generation steps.
3) We can remove the dependency of η on $\lambda_{\min}$ and n by using a sequence of dynamic learning rates: $\eta_t = \frac{1}{\sqrt{t+1}}$. Sublinear regret bounds can be obtained for such a modification but we omit the analysis due to space constraints.

VII. ALGORITHM IMPLEMENTATION AND COMPUTATIONAL COMPLEXITY

Recall from Lemma 1 that cumulative loss is a piece-wise constant function. We use this fact to compute the continuous domain integral in (8) efficiently by splitting the function into multiple rectangular areas of nonuniform base and then summing them up, where we do not make any discretisation error but compute the exact value of the integral. In each round t, we increase the number of intervals by at most 1 as we split the interval containing p_t at p_t. After receiving p_t, we thus have $N \leq t+1$ intervals with boundaries given by $p_{[0]} = 0$, $p_{[i]}$, $1 \leq i \leq t$, and $p_{[N]} = 1$. The weight $w_{i,t}$, $i \leq t+1$ of the interval i in round t is then updated based on, 1) the weights in round $t-1$, and 2) the position of the interval with respect to p_t. Note that in lines 12 of HIL-F and HIL-N, we state that the interval containing p_t should be split and in line 14 we state that the weights should be computed, but without giving more details. Below, we present four algorithmic rules that can be used to compute the probability q_t, interval boundaries $\{p_{[i]}\}$ and weights $\{w_{i,t}\}$, which needs to be computed in order. Let j be the interval strictly below p_t and *dup* be a Boolean variable denoting duplicate p_t.

$$(i)\ j \leftarrow \max\{i : p_{[i]} < p_t\}.$$

$$(ii)\ dup \leftarrow FALSE, \text{ if } p_\tau \neq p_t,\ \forall \tau < t,\ TRUE \text{ otherwise.}$$

$$(iii)\ q_t \leftarrow \frac{\sum_{i=1}^{j} w_{i,t}(p_{[i]} - p_{[i-1]}) + w_{j+1,t}(p_t - p_{[i]})}{\sum_{i=1}^{N} w_{i,t}(p_{[i]} - p_{[i-1]})}$$

$$(iv)\ N \leftarrow \begin{cases} N & (dup = TRUE), \\ N+1 & (dup = FALSE). \end{cases}$$

$$(v)\ p_{[i]} \leftarrow \begin{cases} p_{[i]} & i \leq j \text{ or } (dup = TRUE) \\ p_t & i = j+1 \text{ and } (dup = FALSE) \\ p_{[i-1]} & j+1 < i \leq N \text{ and } (dup = FALSE) \end{cases}$$

$$(vi)\ w_{i,t} \leftarrow \begin{cases} w_{i,t-1}e^{-\eta\beta} & p_{[i]} > p_t,\ (dup = TRUE) \\ w_{i-1,t-1}e^{-\eta\beta} & p_{[i]} > p_t,\ (dup = FALSE) \\ w_{i,t-1}e^{-\eta Y_t} & p_{[i]} \leq p_t, \text{HIL-F} \\ w_{i,t-1}e^{-\eta Y_t/\epsilon} & p_{[i]} \leq p_t, Z_t = 1, \text{HIL-N} \\ w_{i,t-1} & p_{[i]} \leq p_t, Z_t = 0, \text{HIL-N.} \end{cases}$$

In every round of computation, we need a certain constant number of additions, multiplications, and comparisons per interval, irrespective of the number of samples already processed. Thus, the computational complexity in each round is in the order of the number of intervals present in that interval. Now consider a set of n input images. In our proposed algorithms, the number of intervals in round t is upper bounded by $t+1$. Thus, the worst-case computational complexity of HIL-F in round t is $O(t)$. Further, when $\lambda_{\min}$ is the minimum difference between any two probabilities, the maximum number of intervals is clearly upper bounded by $1/\lambda_{\min}$, which reduces the complexity to $O(\min\{t, 1/\lambda_{\min}\})$.

Proposition 1: The computational complexity of HIL-F and HIL-N in round t is $O(\min\{t, 1/\lambda_{\min}\})$.

Note that there can be many intervals with lengths larger than $\lambda_{\min}$, and thus the number of intervals can typically be less than $1/\lambda_{\min}$, which reduces the complexity in practice. As discussed earlier, one might approximate $\lambda_{\min}$ to $1/n$ in some datasets, which gives us a complexity of $O(\min\{t, n\})$ in terms of the number of images. Also note that the above complexities are that of round t, and to get the total complexity of the algorithm, one has to sum it overall t.

Finally, we note that there can be datasets where $\lambda_{\min} < 1/n$ and for such cases the complexity from Proposition 1 will be $O(t)$. For instance, this is the case for the MNIST dataset but is not applicable for the Imagenette dataset with $\lambda_{\min} = \frac{1}{256}$. In this regard, we propose a practical modification to the algorithms by limiting the interval size to a minimum of $\Delta_{\min} > \lambda_{\min}$, where $\Delta_{\min}$ is a parameter chosen based on the complexity and cost tradeoffs. One then considers any different probabilities that lie within $\Delta_{\min}$ of each other as duplicates while generating new intervals in line 12 of HIL-F and HIL-N, which further reduces the complexity to $O(\min\{t, 1/\Delta_{\min}\})$. We observed by choosing different values of $\lambda_{\min}$ (including $\frac{1}{n}$) that over a range of values, there is a notable reduction in algorithm runtime, with negligible difference in the expected average costs.

VIII. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed algorithms HIL-F and HIL-N by comparing them against each other as well as further benchmarks. Our evaluation scenario consists of two different classifiers and four different datasets. Firstly, we use 8-bit quantised MobileNet [4], [44], with width parameter 0.25, to classify the Imagenette and Imagewoof datasets [11]. We use 0.25 for the width parameter

as it reduces the number of filters in each layer drastically, and the resulting MobileNet has a size of 0.5 MB, suitable to fit on an IoT sensor. Imagenette and Imagewoof are derived from Imagenet [5] and each contains a mixture of 10 different image classes with approximately 400 images per class. Out of the two, Imagewoof is a tougher dataset for classification as it contains 10 different breeds of dogs. Next, we use the test set of MNIST dataset [13], which contains 10000 images of handwritten digits from 0 through 9. For this dataset, we train a linear classifier (without regulariser), as the S-ML model. We convert the labels into vectors of size 10. For label l, i.e., digit l, we use all zero vectors except in l^{th} location, where the value is 1. After training the classifier, we scale the output to obtain a probability distribution over the 10 labels. The top-1 accuracy we obtain is 86%. Finally, for CIFAR-10 [14], [15], we use a readily available trained CNN [45] with accuracy 84% as the S-ML model. Note that for all the simulations, we invoke the assumption that the L-ML models have accuracy 1.

As explained in Section III, we choose the expected average regret $\frac{1}{n}\mathbb{E}_{\boldsymbol{Y}}[R_n]$ and expected average cost $\frac{1}{n}\mathbb{E}_{\boldsymbol{Y},\pi}[L(\boldsymbol{\theta}, \boldsymbol{Y})]$ as the metrics to compare the performance. Recall that these metrics are upper bounded by 1, which is the maximum cost in a single round. For simplicity, we refer to them by average regret and average cost, respectively. For the simulations, we take 100 randomisations of the input sequence Y and for each of these randomisations we repeat the simulations 100 times. The randomisation is for the statistical convergence across the sequences of $\boldsymbol{Y}$ (*i.e.*, $\mathbb{E}_{\boldsymbol{Y}}[.]$), and the repetitions are for the convergence over the randomised decisions based on q_t made in line 4 of the algorithms (i.e., $\mathbb{E}_{\pi}[.]$). We also checked with higher numbers of randomisations and repetitions and verified that 100×100 iterations are sufficient for statistical convergence. We use η and ϵ from (12) and Lemma 2, unless mentioned otherwise.

We use the following four baseline algorithms (i.e., policies) to compare the performance of HIL-F and HIL-N.

1) **Genie** – a non-causal policy, where only those images that are misclassified by S-ML are offloaded.
2) $\boldsymbol{\theta^*}$ – an optimal fixed-θ policy. We compute this cost by running a brute-force grid search over all θ.
3) **Full offload** – all images are offloaded to the ES.
4) **No offload** – all images are processed locally.

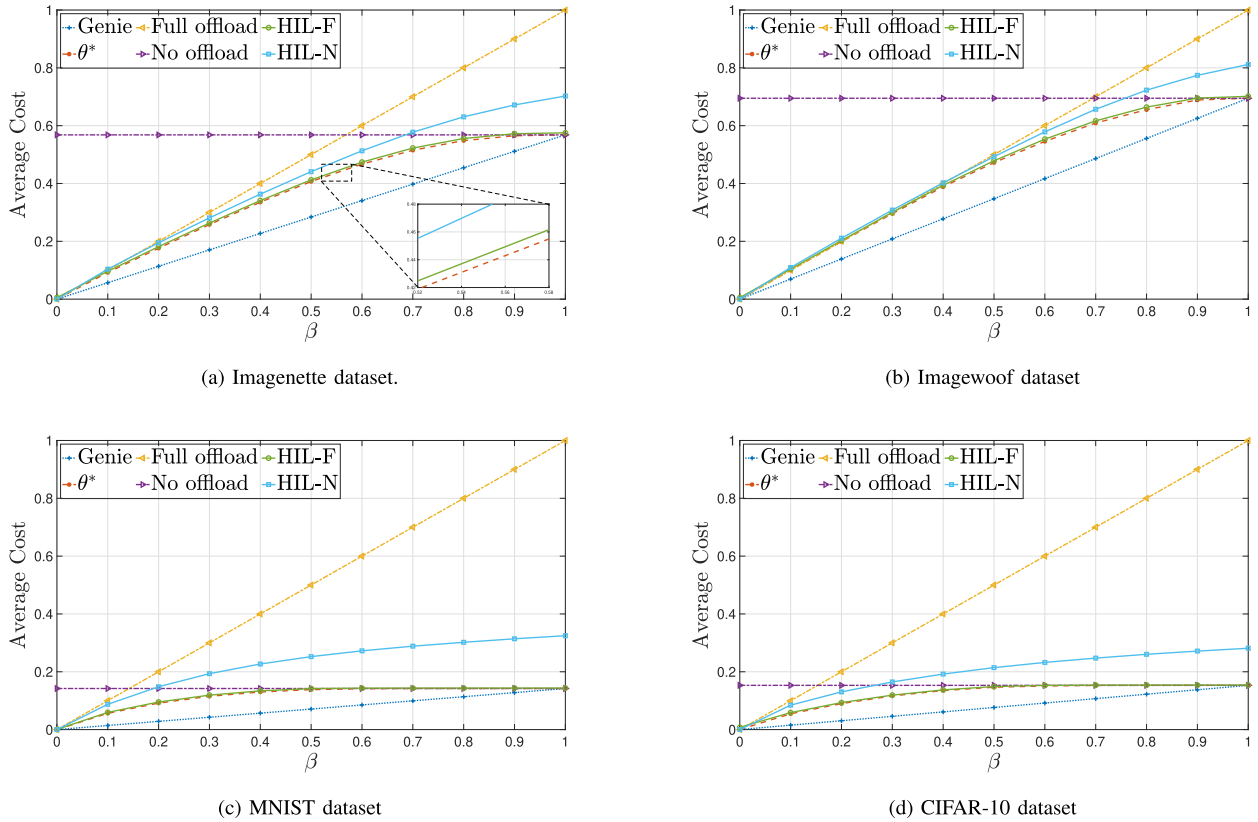
Before we go to the figures, we show the number of images offloaded and the number of images misclassified by different policies for the Imagenette dataset with a total of 3925 images in TABLE 3. These results are basically the data point with $\beta = 0.5$ from Fig. 3a (explained later). We can immediately infer from the table that HIL-F achieves an offloading rate and misclassification rate very close to that of the optimum fixed-θ policy. Further, HIL-F offloads approximately the same number of images as the optimum fixed-θ policy and achieves a top-1 accuracy of 92.3%. Contrast this with a much lower accuracy output of 43.2% by the chosen MobileNet as the S-ML. This also asserts that our framework with the cost structure β and Y indeed facilitates HI by reducing the number of offloaded images that are correctly classified by S-ML. Note that HIL-N also achieves high accuracy 95.2%, but it achieves this at the cost of offloading more images, 18% more than θ^*. This is because HIL-N can only get feedback from L-ML and chooses to offload more images to learn the best threshold. Note that a β of 0.5 corresponds to minimising the total number of errors and offloads and the results can be related to what we can visually infer from Fig. 2 in Section I. The optimum threshold lies around 0.45 which is the minimum threshold above which one can expect to get a correct classification more often than not.

TABLE 3. Number of images offloaded and misclassified for different policies on imagenette with $\beta = 0.5$ and optimal η, ϵ.

Images	Genie	Full offload	No offload	θ^*	HIL-F	HIL-N
offloaded	2230	3925	0	2588	2626	3056
Misclassified	0	0	2230	303	304	191

In Fig. 3, we compare the two proposed algorithms HIL-F and HIL-N with the baselines for all four datasets by plotting the average cost vs. β. Here, Fig. 3a through Fig. 3d correspond to Imagenette, Imagewoof, MNIST, and CIFAR-10 datasets, respectively. Observe that HIL-F performs very close to θ^*, having at most 6% higher total cost than θ^* among all four figures irrespective of the absolute value of the cost or the dataset considered. In Fig. 3a we have also added an inset where we have enlarged a portion of the figure to highlight the distinction between the proposed policies and θ^*. The vertical difference between these two corresponds to the corresponding regret. We can see that HIL-F achieves a cost very close to that of θ^*, having at most 4.5% higher total cost than θ^* throughout the range of β. For instance for the Imagenette dataset with $\beta = 0.5$, this increase is less than 1.4%. HIL-N on the other hand is more sensitive to the properties of the considered dataset. It performs much better than the Full offload policy and also follows a similar trend as that of the HIL-F. However, for larger values of β the comparative performance of HIL-N with the No offload policy deteriorates. This is because even when offloading is not optimum, HIL-N is offloading with a fixed probability $\epsilon > 0$, to learn the ground truth Y. Furthermore, we can see by comparing the four figures that lower the accuracy of S-ML – for instance in Fig. 3b – larger will be the range of β for which HIL-N performs better than both No offload and Full offload policies.

In Fig. 4, we show the dependency of the algorithm on the learning rate parameter η by plotting the average regret obtained by the proposed algorithms vs. the number of images for $\beta = 0.7$ and different values of η. We show the plots for theoretical bound-optimising η, and for HIL-F we also show the plots with a few other η for comparison. First, note that the HIL-N learns slower compared to HIL-F, which is an intuitive behaviour because HIL-N cannot learn from those images that are not offloaded. Also, note that the difference in regret



(a) Imagenette dataset.

(b) Imagewoof dataset

(c) MNIST dataset

(d) CIFAR-10 dataset

FIGURE 3. Average cost incurred by various offloading policies vs. β for different datasets. The bound optimising η and ϵ are used assuming a prior knowledge of $\lambda_{\min}$. Note that the curves corresponding to θ^* and HIL are very close to each other.

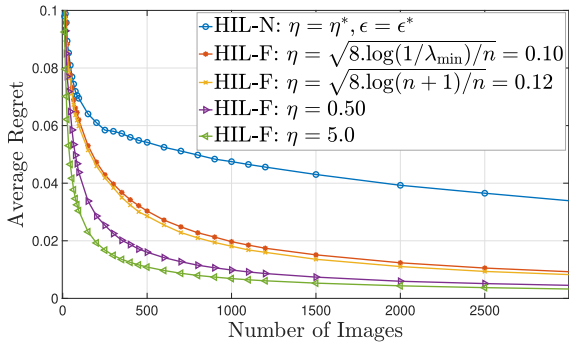


FIGURE 4. Average regret vs. Number of images for $\beta = 0.7$ using HIL-F and HIL-N on the Imagenette database with various η.

incurred by using $\hat{\lambda}_{\min} = 1/(n+1)$ as an approximation of $\lambda_{\min}$ is minimal – in the order 10^{-3}. Recall that the optimum η that we proposed is an optimum for the regret bound, but not necessarily for the regret itself. Hence, it is worth noting that, while using a larger η is slightly beneficial in this particular dataset, this turns out to be deleterious for the regret bound, which is valid for any given dataset. Further, too large an η will give too large weights to the thresholds that achieved lower costs in the past, making the algorithm resemble a deterministic algorithm that cannot guarantee performance [12].

IX. CONCLUSION

We considered an ED embedded with S-ML and an ES having L-ML and explored the idea of HI, where the ED can benefit from only offloading samples for which S-ML outputs incorrect inference. Since an ideal implementation of HI is infeasible, we proposed a novel meta-learning framework where the ED decides to offload or not to offload after observing the maximum probability p in the probability mass function output by S-ML. For the full feedback scenario, we proposed HIL-F, which assigns exponential weights to decision thresholds $\theta \in [0, 1]$ based on past costs and probabilistically chooses a threshold, based on p, to offload or not. For the no-local feedback scenario, we proposed HIL-N, which uses an unbiased estimator of the cost and generates an additional Bernoulli random variable Z and always offloads if $Z = 1$. A novel and unique aspect of the proposed algorithms is that we use non-uniform discretisation, i.e., create new intervals in each round based on p and use these intervals as experts. We proved that HIL-F and HIL-N have sublinear regret bounds $\sqrt{n \ln(1/\lambda_{\min})/2}$ and $O\left(n^{2/3} \ln^{1/3}(1/\lambda_{\min})\right)$, respectively, and have runtime complexity $O\left(\min\{t, 1/\lambda_{\min}\}\right)$ in round t. Here, it is worth noting that the term $1/\lambda_{\min}$ acts similarly to the number of experts in PEA as far as regret bounds are concerned and we have explained simple methods to approximate it. For verifying the results, we generated values of p for four datasets,

namely, Imagenette, Imagewoof, MNIST, and CIFAR-10, and compared the performance of HIL-F and HIL-N with four different baseline policies, including the *fixed-θ* policy. The cost achieved by the proposed algorithms is always lower compared to the *Full offload* and the *No offload* policies and is close to the cost achieved by the optimum fixed-θ policy for a wide range of β. More importantly, the algorithms achieve much higher accuracy compared to S-ML while offloading a marginally higher number of images compared to the optimum fixed-θ policy.

There are multiple directions to which this work can be extended in the future. The major part of the ongoing work is the extension of the algorithm to make a preliminary decision before observing the S-ML output. The envisioned algorithm uses HI as proposed in this work with some minor parameter modifications. Another part of our ongoing work involves modifying the algorithm by replacing certain static parameters with dynamic ones, thereby potentially improving the performance. An example of one such parameter is the learning rate η. We also envision that a future direction where this work would be extended is to consider multiple layers of offload decision, for instance, from device to edge and then edge to cloud.

APPENDIX A
PROOF OF THEOREM 1

We will restate Theorem 1 and prove it.

Theorem 1: For $\eta > 0$, HIL-F achieves the following regret bound:

$$R_n = \bar{L}(\boldsymbol{Y}) - L(\boldsymbol{\theta}^*, \boldsymbol{Y}) \leq \frac{1}{\eta} \ln \frac{1}{\lambda_{\min}} + \frac{n\eta}{8}.$$

Proof: Recall from Lemma 1 that $p_{[i]}$, $B_i = (p_{[i-1]}, p_{[i]}]$, and $l(B_i, Y_t)$ are the i^{th} smallest probability, intervals formed by them, and the constant loss function within that interval at round t, respectively. Also, $\lambda_i = p_{[i]} - p_{[i-1]}$ and $N \leq n+1$ correspond to the length of the intervals i and the total number of intervals, respectively. Finally, $\lambda_{\min} = \min_{1\leq i\leq N} \lambda_i$. Substituting $t = 0$ in (7), we have $W_1 = 1$. Thus, taking logarithm of $\frac{W_{n+1}}{W_1}$ gives,

$$\begin{aligned}
\ln \frac{W_{n+1}}{W_1} &= \ln \int_0^1 e^{-\eta \sum_{t=1}^n l(x,Y_t)}\, \mathrm{d}x \\
&= \ln \sum_{i=1}^N \lambda_i e^{-\eta \sum_{t=1}^n l(B_i,Y_t)} \\
&\geq \ln \max_{1\leq i\leq N} \left(\lambda_{\min} e^{-\eta \sum_{t=1}^n l(B_i,Y_t)}\right) \\
&= -\eta \min_{1\leq i\leq N} \sum_{t=1}^n l(B_i, Y_t) - \ln \frac{1}{\lambda_{\min}} \\
&= -\eta \min_{\theta\in[0,1]} \sum_{t=1}^n l(\theta, Y_t) - \ln \frac{1}{\lambda_{\min}}. \qquad (21)
\end{aligned}$$

Now, we bound the ratio $\frac{W_{t+1}}{W_t}$.

$$\begin{aligned}
\ln\left(\frac{W_{t+1}}{W_t}\right) &= \ln\left(\frac{\int_0^1 w_{t+1}(x)\,\mathrm{d}x}{W_t}\right) \\
&= \ln\left(\int_0^1 \frac{w_t(x)}{W_t} e^{-\eta l(x,Y_t)}\,\mathrm{d}x\right).
\end{aligned}$$

By using Hoeffding's lemma[3] in the above equation, we get

$$\begin{aligned}
&\ln\left(\frac{W_{t+1}}{W_t}\right) \leq -\eta \int_0^1 \frac{w_t(x)}{W_t} l(x, Y_t)\,\mathrm{d}x + \frac{\eta^2}{8} \\
&= -\eta \int_0^{p_t} \frac{w_t(x)}{W_t} l(x, Y_t)\,\mathrm{d}x - \eta \int_{p_t}^1 \frac{w_t(x)}{W_t} l(x, Y_t)\,\mathrm{d}x + \frac{\eta^2}{8} \\
&= -\eta \left(Y_t \int_0^{p_t} \frac{w_t(x)}{W_t}\,\mathrm{d}x + \beta \int_{p_t}^1 \frac{w_t(x)}{W_t}\,\mathrm{d}x\right) + \frac{\eta^2}{8}.
\end{aligned}$$

In the above step, we used (3). Now using (8) to replace the integrals, we get

$$\begin{aligned}
\ln\left(\frac{W_{t+1}}{W_t}\right) &\leq -\eta\left(Y_t q_t + \beta(1-q_t)\right) + \frac{\eta^2}{8} \\
&= -\eta \bar{l}(Y_t) + \frac{\eta^2}{8}. \qquad (22)
\end{aligned}$$

Extending this expression telescopically, we get

$$\begin{aligned}
\ln\left(\frac{W_{n+1}}{W_1}\right) &= \ln\left(\prod_{t=1}^n \frac{W_{t+1}}{W_t}\right) = \sum_{t=1}^n \ln \frac{W_{t+1}}{W_t} \\
&\leq \sum_{t=1}^n \left[-\eta \bar{l}(Y_t) + \frac{\eta^2}{8}\right] \\
&= -\eta \sum_{t=1}^n \bar{l}(Y_t) + \frac{n\eta^2}{8}. \qquad (23)
\end{aligned}$$

Using (21) and (23), we obtain

$$\begin{aligned}
&-\eta \min_{\theta\in[0,1]} \sum_{t=1}^n l(\theta, Y_t) - \ln \frac{1}{\lambda_{\min}} \leq -\eta \sum_{t=1}^n \bar{l}(Y_t) + \frac{n\eta^2}{8} \\
&\Rightarrow \bar{L}(\boldsymbol{Y}) \leq L(\boldsymbol{\theta}^*, \boldsymbol{Y}) + \frac{1}{\eta} \ln \frac{1}{\lambda_{\min}} + \frac{n\eta}{8} \\
&\Rightarrow R_n \leq \frac{1}{\eta} \ln \frac{1}{\lambda_{\min}} + \frac{n\eta}{8}.
\end{aligned}$$

In the last two steps above, we rearranged the terms and divided them with η. ■

APPENDIX B
PROOF OF THEOREM 2

We will restate Theorem 2 and prove it.

Theorem 2: For $\eta, \epsilon > 0$, HIL-N achieves the regret bound

$$R_n \leq n\beta\epsilon + \frac{n\eta}{2\epsilon} + \frac{1}{\eta} \ln(1/\lambda_{\min}). \qquad (24)$$

[3]For a bounded random variable $X \in [a, b]$, Hoeffding's lemma states that $\ln(\mathbb{E}[e^{sX}]) \leq s\mathbb{E}[X] + \frac{s^2(b-a)^2}{8}$.

Proof: **Step 1:** Since the costs incurred and the loss function used for updating the weights are different under HIL-N, we first find a bound for the difference between the expected total cost received and the expected total cost obtained using $\tilde{l}(\theta_t, Y_t)$. From Algorithm 2, we infer that sample t is offloaded if $Q_t = 0$ or $Q_t = 1$ and $Z_t = 1$, and it is not offloaded only when $Q_t = 0$ and $Z_t = 0$. Therefore, we have

$$\mathbb{E}_{Q_tZ}\left[l(\theta_t, Y_t)\right] = \beta[1 - q_t + q_t\epsilon] + q_t(1-\epsilon)Y_t. \quad (25)$$

From (13), we have

$$\tilde{l}(\theta_t, Y_t) = \frac{Y_t}{\epsilon}\,\mathbb{1}(\theta_t \le p_t)\,\mathbb{1}(Z_t = 1) + \beta\,\mathbb{1}(\theta_t > p_t)$$
$$\Rightarrow \mathbb{E}_{Q_tZ}\left[\tilde{l}(\theta_t, Y_t)\right] = Y_t q_t + \beta(1 - q_t). \quad (26)$$

From (25) and (26), we obtain

$$\begin{aligned}
&\mathbb{E}_{Q_tZ}\left[l(\theta_t, Y_t)\right] - \mathbb{E}_{Q_tZ}\left[\tilde{l}(\theta_t, Y_t)\right] \\
&= \beta\epsilon q_t - Y_t\epsilon q_t. \\
&\Rightarrow \mathbb{E}_{\boldsymbol{Q}Z}\left[L(\boldsymbol{\theta}, \boldsymbol{Y})\right] - \sum_{t=1}^{n}\mathbb{E}_{\boldsymbol{Q}Z}\left[\tilde{l}(\theta_t, Y_t)\right] \\
&= \beta\epsilon\sum_{t=1}^{n} q_t - \epsilon\sum_{t=1}^{n} Y_t q_t \\
&\le n\beta\epsilon - \epsilon\sum_{t=1}^{n} Y_t q_t \\
&\Rightarrow -\sum_{t=1}^{n}\mathbb{E}_{\boldsymbol{Q}Z}\left[\tilde{l}(\theta_t, Y_t)\right] \le -\mathbb{E}_{\boldsymbol{Q}Z}\left[L(\boldsymbol{\theta}, \boldsymbol{Y})\right] + n\beta\epsilon. \quad (27)
\end{aligned}$$

In the last step above, we have used $q_t \le 1$, for all t.

Step 2: Using the same analysis to derive (21), we obtain

$$\ln\left(\frac{W_{n+1}}{W_1}\right) \ge -\eta\min_{\theta\in[0,1]}\sum_{t=1}^{n}\tilde{l}(\theta, Y_t) - \ln\frac{1}{\lambda_{\min}}$$

Note that, here we have $\tilde{l}(\theta, Y_t)$ instead of $l(\theta, Y_t)$. Now, using the fact that the expectation over the minimum is upper bounded by the minimum over expectation, we get

$$\begin{aligned}
&\Rightarrow \mathbb{E}_Z\left[\ln\left(\frac{W_{n+1}}{W_1}\right)\right] \ge -\eta\min_{\theta\in[0,1]}\sum_{t=1}^{n}\mathbb{E}_Z\left[\tilde{l}(\theta, Y_t)\right] - \ln\tfrac{1}{\lambda_{\min}} \\
&\Rightarrow \mathbb{E}_Z\left[\ln\left(\frac{W_{n+1}}{W_1}\right)\right] \ge -\eta L(\boldsymbol{\theta}^*, \boldsymbol{Y}) - \ln\tfrac{1}{\lambda_{\min}}. \quad (28)
\end{aligned}$$

Step 3: In the following we find a bound for $\ln(\frac{W_{t+1}}{W_t})$.

$$\begin{aligned}
&\ln\left(\frac{W_{t+1}}{W_t}\right) \\
&= \ln\left(\frac{\int_0^1 w_{t+1}(x)\,\mathrm{d}x}{W_t}\right) \\
&= \ln\left(\int_0^1 \frac{w_t(x)}{W_t}e^{-\eta\tilde{l}(x,Y_t)}\,\mathrm{d}x\right) \quad \text{(using (14))} \\
&\le \ln\left(\int_0^1 \frac{w_t(x)}{W_t}\left(1 - \eta\tilde{l}(x, Y_t) + \frac{\eta^2}{2}\tilde{l}(x, Y_t)^2\right)\mathrm{d}x\right).
\end{aligned}$$

In the above step, we used the fact that $e^{-x} \le 1 - x + x^2/2$. Rearranging the terms, we get

$$\begin{aligned}
&\ln\left(\frac{W_{t+1}}{W_t}\right) \\
&= \ln\left(1 + \int_0^1 \frac{w_t(x)}{W_t}\left(-\eta\tilde{l}(x, Y_t) + \frac{\eta^2}{2}\tilde{l}(x, Y_t)^2\right)\mathrm{d}x\right) \\
&\le \int_0^1 \frac{w_t(x)}{W_t}\left(-\eta\tilde{l}(x, Y_t) + \frac{\eta^2}{2}\tilde{l}(x, Y_t)^2\right)\mathrm{d}x.
\end{aligned}$$

The above step follows from the fact that $\ln(1+x) \le x,\ \forall x > -1$.

$$\Rightarrow \ln\left(\tfrac{W_{t+1}}{W_t}\right) \le \int_0^1 \tfrac{w_t(x)}{W_t}\left(-\eta\tilde{l}(x, Y_t) + \tfrac{\eta^2}{2\epsilon}\tilde{l}(x, Y_t)\right)\mathrm{d}x. \quad (29)$$

In the last step, we have used the fact that $\tilde{l}(x, Y_t) \in [0, 1/\epsilon]$. Note that the integral above can be rearranged as follows:

$$\begin{aligned}
&\int_0^1 \frac{w_t(x)}{W_t}\tilde{l}(x, Y_t)\,\mathrm{d}x \\
&= \int_0^{p_t} \frac{w_t(x)}{W_t}\tilde{l}(x, Y_t)\,\mathrm{d}x + \int_{p_t}^1 \frac{w_t(x)}{W_t}\tilde{l}(x, Y_t)\,\mathrm{d}x \\
&= \frac{Y_t}{\epsilon}\mathbb{1}(Z_t = 1)q_t + \beta(1 - q_t).
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\mathbb{E}_Z\left[\int_0^1 \frac{w_t(x)}{W_t}\tilde{l}(x, Y_t)\,\mathrm{d}x\right] &= Y_t q_t + \beta(1 - q_t) \\
&= \mathbb{E}_{Q_tZ}\left[\tilde{l}(\theta_t, Y_t)\right], \quad (30)
\end{aligned}$$

where we have used (26). Taking expectation with respect Z on both sides in (29) and then substituting (30),

$$\begin{aligned}
&\mathbb{E}_Z\left[\ln\left(\frac{W_{t+1}}{W_t}\right)\right] \\
&\le -\eta\mathbb{E}_{Q_tZ}\left[\tilde{l}(\theta_t, Y_t)\right] + \frac{\eta^2}{2\epsilon}\mathbb{E}_{Q_tZ}\left[\tilde{l}(\theta_t, Y_t)\right] \\
&\le -\eta\mathbb{E}_{Q_tZ}\left[\tilde{l}(\theta_t, Y_t)\right] + \frac{\eta^2}{2\epsilon}. \quad (31)
\end{aligned}$$

Above, we used the fact that $\mathbb{E}_{\boldsymbol{Q}Z}[\tilde{l}(\theta_t, Y_t)] \le 1$. Taking summation of (31) over t, we obtain

$$\begin{aligned}
\mathbb{E}_Z\left[\ln\prod_{t=1}^{n}\left(\frac{W_{t+1}}{W_t}\right)\right] &\le -\eta\sum_{t=1}^{n}\mathbb{E}_{Q_tZ}\left[\tilde{l}(\theta_t, Y_t)\right] + \frac{n\eta^2}{2\epsilon} \\
\Rightarrow \mathbb{E}_Z\left[\ln\left(\frac{W_{n+1}}{W_1}\right)\right] &\le -\eta\left(\mathbb{E}_{\boldsymbol{Q}Z}\left[L(\boldsymbol{\theta}, \boldsymbol{Y})\right] - n\beta\epsilon\right) + \frac{n\eta^2}{2\epsilon}. \quad (32)
\end{aligned}$$

In the last step above, we have used (27). Combining (32) and (28) and rearranging the terms, we obtain

$$\mathbb{E}_{\boldsymbol{Q}Z}\left[L(\boldsymbol{\theta}, \boldsymbol{Y})\right] - L(\boldsymbol{\theta}^*, \boldsymbol{Y}) \le n\beta\epsilon + \frac{n\eta}{2\epsilon} + \frac{1}{\eta}\ln(1/\lambda_{\min}),$$

which is the regret R_n for HIL-N given by (18). ■

APPENDIX C
PROOF OF LEMMA 2

We will now restate Lemma 2 and prove it.

Lemma 2: The function $g(\epsilon, \eta)$ defined in (20) has a global minimum at (ϵ^*, η^*), where $\eta^* = \left(\frac{2\ln^2(1/\lambda_{\min})}{\beta n^2}\right)^{1/3}$ and $\epsilon^* = \sqrt{\frac{\eta}{2\beta}}$. At this minimum, we have,

$$g(\epsilon^*, \eta^*) = 3n^{2/3}\left(\frac{\beta \ln(1/\lambda_{\min})}{2}\right)^{1/3}.$$

Proof: We can easily see the strict convexity of $g(\epsilon, \eta)$ in each dimension ϵ and η independently, which tells us that any inflection point of the function will be either a saddle point or a minima but not a maxima. We equate the first-order partial derivatives to zero to get a set of points given by the equations

$$\frac{\partial g}{\partial \epsilon} = 0 \Rightarrow \epsilon = \sqrt{\frac{\eta}{2\beta}}, \tag{33}$$

$$\frac{\partial g}{\partial \eta} = 0 \Rightarrow \eta = \sqrt{\frac{2\epsilon \ln(1/\lambda_{\min})}{n}}. \tag{34}$$

However, it still remains to check if this point is unique and if this point is indeed a minimum, but not a saddle point. Seeing the uniqueness is straightforward by noting that these two expressions correspond to two non-decreasing, invertible curves in the ϵ–η plane, and thus they have a unique intersection. We find this intersection denoted using (ϵ^*, η^*) by substituting (33) in (34). We obtain

$$\eta^* = \sqrt{\frac{2\epsilon^* \ln(1/\lambda_{\min})}{n}} = \sqrt{\frac{2\sqrt{\eta^*/2\beta}\ln(1/\lambda_{\min})}{n}}.$$

We get η^* and ϵ^* by simplifying the above equation and then substituting it back in (33). Finally, to prove that (ϵ^*, η^*) is indeed a minimum, we verified that the determinant of the Hessian at (ϵ^*, η^*) is positive, the steps of which are not presented due to space constraints. Since (ϵ^*, η^*) is a unique minimum, it should be the global minimum. The proof is complete by substituting (ϵ^*, η^*) in (20). ∎

APPENDIX D
PROOF OF COROLLARY 2

We will now restate Lemma 2 and prove it.

Corollary 2: With $\eta = \left(\frac{2\ln^2(1/\lambda_{\min})}{\beta n^2}\right)^{1/3}$ and $\epsilon = \min\{1, \sqrt{\frac{\eta}{2\beta}}\}$, HIL-N achieves a regret bound sublinear in n:

$$R_n \le 3n^{2/3}\left(\frac{\beta \ln(1/\lambda_{\min})}{2}\right)^{1/3}$$

Proof: Note that, if $\sqrt{\frac{\eta}{2\beta}} \le 1$, then $\epsilon = \sqrt{\frac{\eta}{2\beta}}$ and the results directly follows from Lemma 2. If $\sqrt{\frac{\eta}{2\beta}} > 1$, then we have $\epsilon = 1$. Substituting η value in $\sqrt{\frac{\eta}{2\beta}} > 1$, we obtain

$$\beta < \sqrt{\frac{\sqrt{2}\ln(1/\lambda_{\min})}{n}}. \tag{35}$$

Since $\epsilon = 1$, we will have $Z_t = 1$ for all t, i.e., HIL-N will always offload. Therefore, in this case, the total cost incurred by HIL-N is equal to $n\beta$. Now, using (35), we obtain

$$n\beta < \sqrt{\sqrt{2}\ln(1/\lambda_{\min})/n} = \sqrt{\sqrt{2}n\ln(1/\lambda_{\min})}.$$

Thus, when (35) holds and we have $\epsilon = 1$, the total cost itself is $O(n^{\frac{1}{2}})$ and therefore regret cannot be greater than $O(n^{\frac{1}{2}})$. The result follows by noting that $O(n^{\frac{2}{3}})$ is the larger bound. ∎

APPENDIX E
IMPERFECT L-ML (ACCURACY < 100 %)

In this appendix, we remove the assumption of a perfect L-ML and retrace the steps of the original analysis by considering an additional cost of incorrect inferences at the ES. Let γ be the normalised cost of incorrect inference at the L-ML.[4] Here, normalisation is done according to the steps carried out to force the 0, 1 costs for S-ML inference detailed in Section III; cf. (1). That is, if C_γ is the absolute cost of L-ML inaccuracy, $\gamma = \frac{C_\gamma - C_0}{C_1 - C_0}$. Similar to Y_t defined in Section III, define a random variable X_t that takes value 0 or γ depending on whether the L-ML inference is correct or not. Let $\boldsymbol{X}_t = \{X_\tau\}$, $\tau = 1, 2, \ldots, t \le n$ and $\boldsymbol{X} := \boldsymbol{X}_n$. We make modifications in the definitions to include this random variable to get

$$l(\theta_t, Y_t, X_t) = \begin{cases} Y_t & p_t \ge \theta_t, \\ \beta + X_t & p_t < \theta_t. \end{cases}$$

$$\theta^* = \operatorname*{argmin}_{\theta\in[0,1]} \sum_{t=1}^{n} l(\theta, Y_t, X_t)$$

$$L(\boldsymbol{\theta}^*, \boldsymbol{Y}, \boldsymbol{X}) = \sum_{t=1}^{n} l(\theta^*, Y_t, X_t)$$

With this modification, we define the modified regret, where an additional expectation is taken over the set of L-ML inferences $\boldsymbol{X}$. That is,

$$R_n = \mathbb{E}_{\pi,\boldsymbol{X}}\left[L(\boldsymbol{\theta}, \boldsymbol{Y}, \boldsymbol{X})\right] - L(\boldsymbol{\theta}^*, \boldsymbol{Y}, \boldsymbol{X}).$$

A. CHANGES IN LEMMA 1

With this modification, Lemma 1 follows in a very similar way up to (10). The cost for all θ within an interval B_i takes a constant value of $l(B_i, Y_t, X_t)$, which depends on whether $p_{[i]}$ is greater than p_t or not. We get

$$L(\boldsymbol{\theta}^*, \boldsymbol{Y}, \boldsymbol{X}) = \min_{1\le i\le N}\left\{\beta\sum_{j=1}^{i-1} m_j + \sum_{k=1}^{n}\left(X_{[k]}\mathbb{I}_i + Y_{[k]}\bar{\mathbb{I}}_i\right)\right\},$$

where $\mathbb{I}_i$ and $\bar{\mathbb{I}}_i$ are the short notations of the indicator random variables $\mathbb{I}_{k\le\sum_{j=1}^{i-1} m_j}$ and $\mathbb{I}_{k>\sum_{j=1}^{i-1} m_j}$, respectively. When there

[4] $\gamma = 1$ for an application that does not differentiate between the errors made by S-ML and L-ML. In this case $X_t \in \{0, 1\}$.

are no repetitions, we get the following as the counterpart result of Lemma 1.

$$L(\boldsymbol{\theta}^*, \boldsymbol{Y}, \boldsymbol{X}) = \min_{1\le i\le n+1}\left\{(i-1)\beta + \sum_{k=1}^{i-1} X_{[k]} + \sum_{k=i}^{n} Y_{[k]}\right\}. \tag{36}$$

B. CHANGES IN THEOREM 1

Recall that we actually do not use the expression in (36) in the proof of regret bounds, but rather use it to assert that the loss function is piece-wise constant within the intervals created by p_t. To derive the regret bound, we modify $\bar{l}(Y_t)$ and define $\bar{l}(Y_t, X_t)$ as

$$\begin{aligned}\bar{l}(Y_t, X_t) = \mathbb{E}_{Q_t}[l(\theta_t, Y_t, X_t)] = Y_t q_t + (\beta + X_t)(1-q_t)\\ \Rightarrow \mathbb{E}_{X_t}\left[\bar{l}(Y_t, X_t)\right] = Y_t q_t + (\beta + \mathbb{E}[X_t])(1-q_t)\end{aligned}$$

Similar changes apply to the cumulative loss function $\bar{L}(Y_t, X_t)$ as well. Note that, $\mathbb{E}_{X_t}\left[\bar{l}(Y_t, X_t)\right]$ is equivalent to the $\bar{l}(Y_t)$ in the original problem with a different offload cost $\beta' = \beta + \mathbb{E}[X_t]$. They are the same when $\mathbb{E}[X_t] = 0$, or in other words, the L-ML inference is perfect. Given these modifications, the algorithms and the analysis follow with one minor caveat: β' can be above 1 even if the offload cost is less than the cost for incorrect S-ML inference and in such cases, a trivial decision of always choosing the S-ML inference needs to be taken. For example, assuming that the misclassification ratio δ of the L-ML is known, $\beta' = \beta + \delta\gamma$, $\forall t$. Then the trivial decision to not offload is made when $\delta > \frac{1-\beta}{\gamma} = \frac{C_1 - C_\beta}{C_\gamma - C_0}$.

Now, the analysis of Theorem 1 can be carried out in a similar fashion with β replaced by $\beta + X_t$. Some of the important steps are given below, where (37) to (39) corresponds to (21) to (23) from Section V.

$$\ln\frac{W_{n+1}}{W_1} \ge -\eta \min_{\theta\in[0,1]}\sum_{t=1}^{n} l(\theta, Y_t, X_t) - \ln\frac{1}{\lambda_{\min}}. \tag{37}$$

$$\begin{aligned}\ln\left(\frac{W_{t+1}}{W_t}\right) &\le -\eta\left(Y_t q_t + (\beta + X_t)(1-q_t)\right) + \frac{\eta^2}{8}\\ &= -\eta\bar{l}(Y_t, X_t) + \frac{\eta^2}{8}.\end{aligned} \tag{38}$$

Extending this expression telescopically,

$$\begin{aligned}\ln\left(\frac{W_{n+1}}{W_1}\right) &\le \sum_{t=1}^{n}\left(-\eta\bar{l}(Y_t, X_t) + \frac{\eta^2}{8}\right)\\ &= -\eta\sum_{t=1}^{n}\bar{l}(Y_t, X_t) + \frac{n\eta^2}{8}.\end{aligned} \tag{39}$$

Combining (37) and (39), we get

$$\bar{L}(\boldsymbol{Y}, \boldsymbol{X}) \le L(\boldsymbol{\theta}^*, \boldsymbol{Y}, \boldsymbol{X}) + \frac{1}{\eta}\ln\frac{1}{\lambda_{\min}} + \frac{n\eta}{8}.$$

Taking expectations with respect to $\boldsymbol{X}$ (which are i.i.d.), we get

$$\begin{aligned}\mathbb{E}_{\boldsymbol{X}}\left[\bar{L}(\boldsymbol{Y}, \boldsymbol{X})\right] &\le \mathbb{E}_{\boldsymbol{X}}\left[L(\boldsymbol{\theta}^*, \boldsymbol{Y}, \boldsymbol{X})\right] + \frac{1}{\eta}\ln\frac{1}{\lambda_{\min}} + \frac{n\eta}{8}\\ \Rightarrow R_n &\le \frac{1}{\eta}\ln\frac{1}{\lambda_{\min}} + \frac{n\eta}{8}.\end{aligned} \tag{40}$$

Here, the regret is an expectation over the L-ML inferences, which are assumed to be i.i.d. and carried out without any information about the S-ML inference.

C. CHANGES IN THEOREM 2

As before, we redo Theorem 2 by simply substituting $\beta + X_t$ instead of β in the loss function at time t. Let the pseudo loss function $\tilde{l}(\theta_t, Y_t, X_t)$ be defined as follows:

$$\tilde{l}(\theta_t, Y_t, X_t) = \begin{cases} 0 & p_t \ge \theta_t, Z_t = 0,\\ \dfrac{Y_t}{\epsilon} & p_t \ge \theta_t, Z_t = 1,\\ \beta + X_t & p_t < \theta_t.\end{cases} \tag{41}$$

Facilitated by the fact that X_t does not depend on Q_t, Z or algorithm π and thus $\mathbb{E}_{Q_t Z}[X_t] = X_t$, we can rewrite (25) and (26) respectively as

$$\begin{aligned}\mathbb{E}_{Q_t Z}\left[l(\theta_t, Y_t, X_t)\right] &= (\beta + X_t)[1 - q_t + q_t\epsilon] + q_t(1-\epsilon)Y_t,\\ \mathbb{E}_{Q_t Z}\left[\tilde{l}(\theta_t, Y_t, X_t)\right] &= Y_t q_t + (\beta + X_t)(1-q_t).\end{aligned}$$

Combining the two, we get

$$\begin{aligned}&\mathbb{E}_{Q_t Z}[l(\theta_t, Y_t, X_t)] - \mathbb{E}_{Q_t Z}\left[\tilde{l}(\theta_t, Y_t, X_t)\right]\\ &= (\beta + X_t)\epsilon q_t - Y_t\epsilon q_t.\\ &\Rightarrow \mathbb{E}_{\boldsymbol{Q}Z}[L(\boldsymbol{\theta}, \boldsymbol{Y}, \boldsymbol{X})] - \sum_{t=1}^{n}\mathbb{E}_{\boldsymbol{Q}Z}\left[\tilde{l}(\theta_t, Y_t, X_t)\right]\\ &\le \epsilon\sum_{t=1}^{n}(\beta + X_t) - \epsilon\sum_{t=1}^{n} Y_t q_t\\ &\Rightarrow -\sum_{t=1}^{n}\mathbb{E}_{\boldsymbol{Q}Z}\left[\tilde{l}(\theta_t, Y_t, X_t)\right]\\ &\le -\mathbb{E}_{\boldsymbol{Q}Z}[L(\boldsymbol{\theta}, \boldsymbol{Y}, \boldsymbol{X})] + \epsilon\sum_{t=1}^{n}(\beta + X_t).\end{aligned} \tag{42}$$

Note that the above is the counterpart of (27).

The remainder of the steps to modify (28) through (31) follow similarly, where we get

$$\begin{aligned}&\mathbb{E}_Z\left[\ln\left(\frac{W_{n+1}}{W_1}\right)\right]\\ &\ge -\eta L(\boldsymbol{\theta}^*, \boldsymbol{Y}, \boldsymbol{X}) - \ln\tfrac{1}{\lambda_{\min}},\\ &\mathbb{E}_Z\left[\ln\left(\frac{W_{n+1}}{W_1}\right)\right]\\ &\le -\eta\left(\mathbb{E}_{\boldsymbol{Q}Z}[L(\boldsymbol{\theta}, \boldsymbol{Y}, \boldsymbol{X})] - \epsilon\sum_{t=1}^{n}(\beta + X_t)\right) + \frac{n\eta^2}{2\epsilon}.\end{aligned}$$

Combining the above two, we get,

$$\mathbb{E}_{\boldsymbol{Q}Z}[L(\boldsymbol{\theta}, \boldsymbol{Y}, \boldsymbol{X})] - L(\boldsymbol{\theta}^*, \boldsymbol{Y}, \boldsymbol{X})$$

$$\leq \epsilon \sum_{t=1}^{n} (\beta + X_t) + \frac{n\eta}{2\epsilon} + \frac{1}{\eta} \ln(1/\lambda_{\min}).$$

Taking expectation with respect to $\boldsymbol{X}$, we get

$$R_n \leq n\epsilon(\beta + \delta\gamma) + \frac{n\eta}{2\epsilon} + \frac{1}{\eta} \ln(1/\lambda_{\min}). \tag{43}$$

Remarks:

1) Comparing (11) and (40), we see that for HIL-F, the regret bound is same with or without a perfect L-ML. This could be attributed to the fact that the bound is independent of the offload cost β.
2) The regret for HIL-N in (43) falls back to (19), when $\delta = 0$, that is when the L-ML is perfect.
3) The regret bound with a varying offload cost β_t, $t = 1, 2 \ldots$ can be analysed in a very similar manner. We do this by substituting $\beta + X_t$ in the modified loss function to β_t. The results inherit the similarity with $\beta' = \beta + \mathbb{E}[X_t] = \beta + \delta\gamma$ replaced by $\mathbb{E}[\beta_t]$.

REFERENCES

[1] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[2] R. Sanchez-Iborra and A. F. Skarmeta, "TinyML-enabled frugal smart objects: Challenges and opportunities," *IEEE Circuits Syst. Mag.*, vol. 20, no. 3, pp. 4–18, 3rd Quart., 2020.

[3] L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, "Model compression and hardware acceleration for neural networks: A comprehensive survey," *Proc. IEEE*, vol. 108, no. 4, pp. 485–532, Apr. 2020.

[4] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2009, pp. 248–255.

[6] M. Wortsman, G. Ilharco, S. Gadre, R. Roelofs, R. Gontijo-Lopes, and A. S. Morcos, "Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *Proc. 39th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, 2022, pp. 23965–23998.

[7] Y. Kang et al., "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," in *Proc. 22nd Int. Conf. Architect. Support Program. Lang. Operating Syst.*, 2017, pp. 615–629.

[8] E. Li, L. Zeng, Z. Zhou, and X. Chen, "Edge AI: On-demand accelerating deep neural network inference via edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 447–457, Jan. 2020.

[9] C. Hu and B. Li, "Distributed inference with deep learning models across heterogeneous edge devices," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2022, pp. 330–339.

[10] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "BranchyNet: Fast inference via early exiting from deep neural networks," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Jul. 2016, pp. 2464–2469.

[11] J. Howard and S. Gugger, "Fastai: A layered API for deep learning," *Information*, vol. 11, no. 2, p. 108, 2020. [Online]. Available: https://github.com/fastai/imagenette

[12] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[13] Y. LeCun and C. Cortes. (2010). *MNIST Handwritten Digit Database*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[14] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 0, 2009.

[15] A. Krizhevsky. (2009). *The CIFAR-10 Dataset*. [Online]. Available: https://www.cs.toronto.edu/

[16] G. Al-Atat, A. Fresa, A. P. Behera, V. N. Moothedath, J. Gross, and J. P. Champati, "The case for hierarchical deep learning inference at the network edge," in *Proc. 1st Int. Workshop Netw. AI Syst.*, 2023, pp. 1–6, doi: 10.1145/3597062.3597278.

[17] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct. 2009.

[18] E. Cuervo et al., "MAUI: Making smartphones last longer with code offload," in *Proc. 8th Int. Conf. Mobile Syst., Appl. Services*, 2010, pp. 49–62.

[19] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 2, pp. 319–333, Feb. 2019.

[20] S. Sundar, J. P. Champati, and B. Liang, "Multi-user task offloading to heterogeneous processors with communication delay and budget constraints," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1958–1974, Jul. 2022.

[21] S. S. Ogden and T. Guo, "MDINFERENCE: Balancing inference accuracy and latency for mobile applications," in *Proc. IEEE IC2E*, Apr. 2020, pp. 28–39.

[22] A. Fresa and J. Prakash Champati, "Offloading algorithms for maximizing inference accuracy on edge device under a time constraint," 2021, *arXiv:2112.11413*.

[23] I. Nikoloska and N. Zlatanov, "Data selection scheme for energy efficient supervised learning at IoT nodes," *IEEE Commun. Lett.*, vol. 25, no. 3, pp. 859–863, Mar. 2021.

[24] J. Wang et al., "Bandwidth-efficient live video analytics for drones via edge computing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 159–173.

[25] J. Wang et al., "Edge-based live video analytics for drones," *IEEE Internet Comput.*, vol. 23, no. 4, pp. 27–34, Jul./Aug. 2019.

[26] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 1135–1143.

[27] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Computer Vision—ECCV 2016*, vol. 9908. Cham, Switzerland: Springer, 2016, pp. 525–542.

[28] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, 2014, pp. 1269–1277.

[29] Z. Wang, W. Bao, D. Yuan, L. Ge, N. H. Tran, and A. Y. Zomaya, "SEE: Scheduling early exit for mobile DNN inference during service outage," in *Proc. 22nd Int. ACM Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, Nov. 2019, pp. 279–288.

[30] B. Taylor, V. S. Marco, W. Wolff, Y. Elkhatib, and Z. Wang, "Adaptive deep learning model selection on embedded systems," *ACM SIGPLAN Notices*, vol. 53, no. 6, pp. 31–43, Jun. 2018.

[31] C. Hu, W. Bao, D. Wang, and F. Liu, "Dynamic adaptive DNN surgery for inference acceleration on the edge," in *Proc. IEEE Conf. Comput. Commun.*, Jul. 2019, pp. 1423–1431.

[32] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.

[33] X.-Y. Zhang, G.-S. Xie, X. Li, T. Mei, and C.-L. Liu, "A survey on learning to reject," *Proc. IEEE*, vol. 111, no. 2, pp. 185–215, Feb. 2023.

[34] N. Charoenphakdee, Z. Cui, Y. Zhang, and M. Sugiyama, "Classification with rejection based on cost-sensitive classification," in *Proc. ICML*, in Proceedings of Machine Learning Research, vol. 139, 2021, pp. 1507–1517.

[35] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," 2016, *arXiv:1610.02136*.

[36] C. De Stefano, C. Sansone, and M. Vento, "To reject or not to reject: That is the question-an answer in case of neural classifiers," *IEEE Trans. Syst., Man, Cybern., C, Appl. Rev.*, vol. 30, no. 1, pp. 84–94, Feb. 2000.

[37] L. P. Cordella, C. De Stefano, F. Tortorella, and M. Vento, "A method for improving classification reliability of multilayer perceptrons," *IEEE Trans. Neural Netw.*, vol. 6, no. 5, pp. 1140–1147, Sep. 1995.

[38] P. Auer, R. Ortner, and C. Szepesvári, "Improved rates for the stochastic continuum-armed bandit problem," in *Proc. Int. Conf. Comput. Learn. Theory*. Berlin, Germany: Springer, 2007.

[39] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári, "X-armed bandits," *J. Mach. Learn. Res.*, vol. 12, pp. 1655–1695, Jul. 2011.

[40] S. Singh, "Continuum-armed bandits: A function space perspective," in *Proc. 24th Int. Conf. Artif. Intell. Statist.*, vol. 130, 2021, pp. 2620–2628.

[41] S. Bubeck, "Introduction to online optimization," Dept. Oper. Res. Financial Eng., Princeton Univ., Princeton, NJ, USA, Lect. Notes, 2011, vol. 2, pp. 1–86.
[42] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, no. 3, pp. 427–485, May 1997.
[43] N. Cesa-Bianchi, G. Lugosi, and G. Stoltz, "Minimizing regret with label efficient prediction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 2152–2162, Jun. 2005.
[44] (Aug. 2022). *GitHub Repository, Keras.io*. [Online]. Available: https://keras.io/api/applications/mobilenet/
[45] GitHub Repository. (2021). *Retrieving the Predictions in the CIFAR-10 Dataset*. [Online]. Available: https://github.com/TracyRenee61/CIFAR_10

VISHNU NARAYANAN MOOTHEDATH received the bachelor's degree in electrics and communication engineering from the National Institute of Technology (NIT), Calicut, in 2012, and the master's degree in communication systems from the Indian Institute of Technology (IIT) Madras in 2016. He is currently pursuing the Ph.D. degree with the Department of Intelligent Systems, School of Electrical Engineering and Computer Science (EECS), KTH Royal Institute of Technology. He was with cellular industry, such as Intel India Pvt. Ltd., and Apple India Pvt. Ltd., in their respective LTE/5G-NR Base-Band Modem Group, for five years. His current research is in the area of edge computing and performance optimization, with a specific focus on improving the energy efficiency and responsiveness of edge computing systems through optimized sampling.

JAYA PRAKASH CHAMPATI (Member, IEEE) received the Bachelor of Technology degree from the National Institute of Technology, Warangal, India, in 2008, the Master of Technology degree from the Indian Institute of Technology (IIT) Bombay, India, in 2010, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Canada, in 2017. Prior to joining Ph.D. degree, he was with Broadcom Communications, where he involved in developing the LTE MAC layer. From 2017 to 2020, he was a Postdoctoral Researcher with the Division of Information Science and Engineering, EECS, KTH Royal Institute of Technology, Sweden. He is currently a Research Assistant Professor with the IMDEA Networks Institute, Madrid, Spain. His general research interest is in the design and analysis of algorithms for scheduling problems that arise in networking and information systems. Currently, his research focus is in edge computing/intelligence, age of information, cyber-physical systems (CPS), and the Internet of Things (IoT). He was a recipient of the Best Paper Award at IEEE National Conference on Communications, India, in 2011.

JAMES GROSS (Senior Member, IEEE) received the Ph.D. degree from TU Berlin in 2006. From 2008 to 2012, he was with RWTH Aachen University, as an Assistant Professor and a Research Associate with the RWTH's Center of Excellence on Ultra-High Speed Mobile Information and Communication (UMIC). Since November 2012, he has been with the Electrical Engineering and Computer Science School, KTH Royal Institute of Technology, Stockholm, where he is a Professor of machine-to-machine communications. At KTH, he was the Director of the ACCESS Linnaeus Centre, from 2016 to 2019, while he is currently the Associate Director of the newly formed KTH Digital Futures Research Center, and the Co-Director of the newly formed VINNOVA Competence Center on Trustworthy Edge Computing Systems and Applications (TECoSA). He has authored over 150 (peer-reviewed) papers in international journals and conferences. His research interests are in the area of mobile systems and networks, with a focus on critical machine-to-machine communications, edge computing, resource allocation, and performance evaluation. His work has been awarded multiple times, including the Best Paper Awards at ACM MSWiM 2015, IEEE WoWMoM 2009, and European Wireless 2009. In 2007, he was a recipient of the ITG/KuVS Dissertation Award for the Ph.D. Thesis.