Inference Offloading for Cost-Sensitive Binary Classification at the Edge

Vishnu Narayanan Moothedath[1], Umang Agarwal[2], Umeshraja N[2], James Richard Gross[1], Jaya Prakash Champati[3], Sharayu Moharir[2]

[1]Information Science and Engineering, KTH Royal Institute of Technology, Stockholm, Sweden
[2]Electrical Engineering, Indian Institute of Technology Bombay, Mumbai, India
[3]Computer Science, University of Victoria, Victoria, Canada

Abstract

We investigate a binary classification problem in an edge intelligence system where false negatives are more costly than false positives. The system features a compact, locally deployed model, supplemented by a larger, remote model that is accessible via the network, albeit at an offloading cost. For each sample, our system first uses the locally deployed model for inference. Based on the output of the local model, the sample may be offloaded to the remote model. This work aims to understand the fundamental trade-off between classification accuracy and the offloading costs within such a hierarchical inference (HI) system. To optimise this system, we propose an online learning framework that continuously adapts a pair of thresholds on the local model's confidence scores. These thresholds determine the prediction of the local model and whether a sample is classified locally or offloaded to the remote model. We present a closed-form solution for the setting where the local model is calibrated. For the more general case of uncalibrated models, we introduce H2T2, an online two-threshold hierarchical inference policy, and prove it achieves sublinear regret. H2T2 is model-agnostic, requires no training, and learns during the inference phase using limited feedback. Simulations on real-world datasets show that H2T2 consistently outperforms naive and single-threshold HI policies, sometimes even surpassing single-threshold offline optima. The policy also demonstrates robustness to distribution shifts and adapts effectively to mismatched classifiers.

Code and datasets —
https://github.com/vnmo/H2T2-AAAI-2026
Extended version — https://arxiv.org/abs/2509.15674

1 Introduction

Edge intelligence systems often operate in critical and resource-constrained environments, where inference tasks must carefully balance cost and accuracy. Examples include event detection, typically framed as a binary classification problem with applications such as medical diagnostics, surveillance, and smart cameras detecting theft, where high reliability must be achieved at a low cost. For instance, consider a medical scenario where a lightweight local inference model performs radiology screening of patient scans

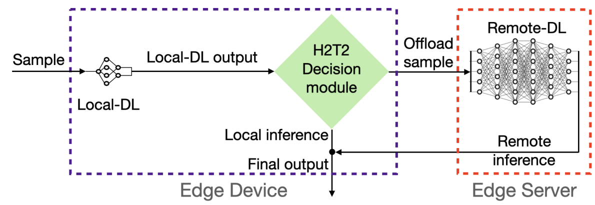
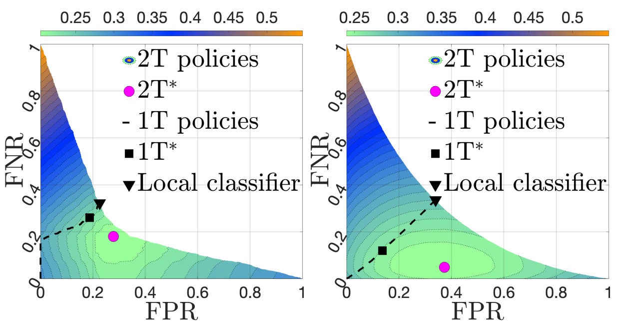


Figure 1: Architecture of the proposed HI-Hedge with Two Thresholds (H2T2) offload decision-making policy.

but may struggle with ambiguous or complex cases. Offloading such samples to a more accurate remote model– or a human specialist– can considerably improve reliability, but at the cost of significant networking and access costs. The consequences of false positives (FP) and false negatives (FN) are typically asymmetric in such settings. For example, missing an incoming threat or a cancer case may be far costlier than a false alarm. Moreover, retraining or updating the local model to adapt to distribution shifts or cost changes may be impractical due to network limitations or model complexity. As a result, training-independent, inference-time decisions become essential. In some systems, pre-training may not be possible at all, requiring models to learn on the fly (Sturzinger and Satyanarayanan 2024).

We consider an edge intelligence system with an edge device and a remote server. The device is equipped with a pretrained local deep learning (LDL) model tasked with cost-sensitive binary classification. For each sample, it can either make a local decision using the LDL output or offload it–at a cost–to a larger, more accurate remote DL (RDL) model. We aim to design an inference-time offloading policy with asymmetric FP and FN costs that selectively offload ambiguous samples based on the LDL output. We use Hierarchical Inference (HI) (Moothedath, Champati, and Gross 2024; Beytur et al. 2024; Al-Atat et al. 2024), a post-hoc meta learning framework where each sample is first sent to the LDL, and the decision to offload to the RDL is made based on the LDL's output; see Figure 1. Prior HI works learned a threshold and offloads a sample if the LDL's confidence (e.g., maximum softmax value) of its decision falls below it.



(a) BreakHis (b) Synthetic

Figure 2: FPR vs. FNR and average cost of 1T (single-threshold) and 2T (two-threshold) policies on (a) BreakHis, and (b) synthetic configurations from TABLE 2. Normalised costs of false positive, false negative, and offload are 0.7, 1, and 0.3, respectively. Optima of IT and 2T polcies are denoted using 1T* and 2T*, respectively.

Related work category	AA	AC	IT	DMA
Rejection(L2R), Deferral(L2D)	✓	×	×	×
Cost-Sensitive Classification	×	✓	×	×
HI–Single Threshold	✓	×	✓	✓
H2T2 (proposed)	✓	✓	✓	✓

Table 1: Comparison of the proposed policy with related work categories across key properties: ambiguity-aware (AA), asymmetric costs (AC), inference-time learning (ITL), and dataset & model agnostic (DMA).

Related Works

HI has received considerable attention (Nikoloska and Zlatanov 2021; Letsiou et al. 2024; Al-Atat et al. 2023; Moothedath, Champati, and Gross 2024; Beytur et al. 2024; Al-Atat et al. 2024; Behera et al. 2025; Datta, Moharir, and Champati 2025), particularly within the edge computing community. Nikoloska and Zlatanov; Al-Atat et al.; Behera et al. assumed a fixed offloading cost per sample and used a predetermined threshold–computed from the training data–on the maximum softmax value of the LDL to make the offloading decision. To handle the varying offloading costs and distribution drifts between training and inference phases, Moothedath, Champati, and Gross; Beytur et al.; Al-Atat et al.; Letsiou et al. investigated learning the optimum threshold online. Moothedath, Champati, and Gross formulated HI as a variant of the classical Prediction with Expert Advice (PEA) (Cesa-Bianchi and Lugosi 2006) and provided dataset-dependent regret bounds. Al-Atat et al. provided improved regret bounds for the problem, considering a more general setting with random and potentially correlated offloading costs. Beytur et al. extended HI to multiple devices and studied maximising accuracy subject to an average offloading cost constraint. In contrast to existing work on HI, we consider asymmetric costs for FPs and FNs.

Our work is also related to the Learning to Reject (L2R) and Learning to Defer (L2D) works by (Cortes, DeSalvo, and Mohri 2016; Saerens, Latinne, and Decaestecker 2002; Ramaswamy, Tewari, and Agarwal 2018; Ni et al. 2019; Narasimhan et al. 2022), which focus on training models with an option to reject, or defer to an expert model–at some cost. In contrast, our meta-learning framework is proposed for the inference phase, where the pre-trained LDL remains unaltered, and it does not require offline RDL inferences. This enables the use of off-the-shelf models on both the edge device and the server. In contrast to cost-sensitive classification (Elkan 2001), our problem requires ambiguity-awareness not only for cost-sensitive classification but also to make the offloading decision. Finally, our framework handles Out-of-Distribution (OOD) samples at inference time, allows independent costs for local processing, offloading, false positives, and false negatives, and allows the offloading cost to be random and unknown a priori. We summarise the difference with related works in Table 1.

While prior HI works utilise the LDL output to make naive *argmax*-based LDL inference when not offloading, we use it to make our own cost-sensitive decisions–serving as both an offloading and a local inference policy. More importantly, all HI works use a single threshold for offloading decisions. We argue that, for cost-sensitive classification, using a single threshold yields sub-optimal solutions. To illustrate this, in Fig. 2 we use two dataset-classifier combinations and show the false-positive rates (FPRs), false negative rates (FNRs), and the average costs achievable by single-threshold and two-threshold policies. While single-threshold policies can reduce FPR and FNR, even the best single-threshold policy cannot achieve the lowest possible overall cost. Instead, we propose a two-threshold HI decision rule and present a sublinear regret algorithm to solve the problem at hand.

Our Contributions

Our key contributions are outlined as follows:

- For calibrated LDL models, we derive a two-threshold rule for the softmax value that minimises the Bayesian expected cost. When FP and FN costs are equal and the offloading cost is constant, this rule reduces to Chow's rule for classification with rejection (Chow 2003).
- Motivated by this, we propose a two-threshold-based HI policy called H2T2 to solve the problem and show that it has $O(T^{\frac{2}{3}})$ regret. Unlike the standard PEA, ground truth labels are unavailable during inference, and H2T2 achieves this sublinear regret by estimating local misclassifications via offloading a fraction of input samples. Experiments on real-world and synthetic datasets demonstrate that H2T2 incurs lower costs compared to the baseline policies, and in some cases, lower than the offline single-threshold optima. Furthermore, we demonstrate that H2T2 cost gains are significant for OOD data. While some cost regions favour naive policies, these regions are unknown a priori.

In the following sections, we present the problem setup, theoretical results for calibrated models, the proposed algorithm with regret analysis, and experimental evaluation. Proofs are provided in the appendix.

2 Problem Setting

We consider a system consisting of an edge device and an edge server. The device must infer relevant information from each sample that arrives periodically, which is then used to make control decisions. This work focuses on the case where this inference task is a cost-sensitive binary classification or event detection. The system is equipped with two deep learning (DL) models trained for this task: a local DL (LDL) model deployed on the device and a remote DL (RDL) model deployed on the server. Given the memory and computing constraints of the device, the LDL is lightweight and, therefore, has lower accuracy than the RDL. We focus on the HI framework, where each sample is first processed by the LDL. Based on its output, the system either makes a local inference or offloads the sample to the RDL for better inference.

Inference Task Assume samples x_t arrive i.i.d. in time slots indexed by t. The goal of the classifier is to map each sample to one of two possible classes, 0 and 1. To this end, the model outputs a softmax tuple $(f_0(x_t), f_1(x_t)) \in \mathbb{R}^2$ with $0 \le f_0(x_t), f_1(x_t) \le 1$, and $f_0(x_t) + f_1(x_t) = 1$. In the remainder of this work, we define the class 1 as the class or event of interest and $f_t := f_1(x_t)$ as the corresponding LDL output. We assume that the model outputs are expressed using b bits and therefore are quantised into 2^b values. This is true for any practical system and has also been assumed in related literature.

Loss Function We denote the LDL and RDL models by the binary classifiers $h_l(.)$ and $h_r(.)$. For each sample x_t, we either use $h_l(x_t)$ or offload the sample and use $h_r(x_t)$ as our inference. In the absence of knowledge of the ground truth class for each sample, we treat the inference of the RDL as a proxy for this ground truth. This is justified by the fact that in any non-trivial scenario, the LDL aims to achieve the performance of the RDL, and no more. Further, this treatment adds simplicity and readability, and extensions to having an imperfect RDL can be done by following the same approach that is used for the single threshold HI in Moothedath, Champati, and Gross. Under this consideration, we have two types of classification errors that can occur when the inference is made locally:

– False-positive (FP), when, $h_l(x_t) = 1$ and $h_r(x_t) = 0$,

– False-negative (FN), when, $h_l(x_t) = 0$ and $h_r(x_t) = 1$.

We use δ_1, δ_{-1} and β_t to denote the normalised false positive, false negative, and offload cost at time t. Here, $\beta_t < \beta$ is generated by an oblivious adversary, and is available at least after round t. Normalisation is achieved by subtracting the common LDL processing cost and scaling by the maximum cost, thereby mapping all values to the range $[0, 1]$. When a local inference is made, the system incurs a loss of ϕ_t:

$$\phi_t = \begin{cases} \delta_1 & \text{if FP} \\ \delta_{-1} & \text{if FN} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

We assume that β_t is presented to the policy at the start of round t. The system is trivial when $\beta_t \ge \max\{\delta_1, \delta_{-1}\}$, enabling an assumption of $\beta_t \le \beta = 1$.

Let π denote the policy (or algorithm) making offloading decisions and $l_t(\pi)$ denote the incurred true loss of this policy at round t. It follows that

$$l_t(\pi) = \begin{cases} \phi_t & \text{if } h_l(x_t) \text{ is used} \\ \beta_t & \text{otherwise.} \end{cases} \tag{2}$$

The corresponding cumulative loss $L_T(\pi) = \sum_{t=1}^{T} l_t(\pi)$. In the remainder of this work, we omit the argument π when it is clear from the context.

Threshold-based Policies and Performance Metric

Recall that for each sample, LDL outputs a score f_t corresponding to class 1. It follows that if f_t is large (close to 1) or small (close to 0), it is very likely that the ground truth is 1 or 0. On the other hand, intermediate values of f_t imply ambiguity. Motivated by this, we focus on threshold-based HI policies where the offloading decision in each slot t is governed by a tuple $\vec{\theta} := (\theta_l, \theta_u)$, where $0 \le \theta_l \le \theta_u \le 1$. A sample is offloaded if $\theta_l \le f_t < \theta_u$, predicted as class 1 if $f_t \ge \theta_u$, and class 0 otherwise. Our goal in this work is to learn these thresholds online.

We say that a policy is a fixed-threshold (fixed-$\vec{\theta}$) policy if $\vec{\theta}$ does not change over the time-horizon of interest. Let the losses associated with this threshold-based policy that uses $\vec{\theta}$ as the threshold tuple be denoted by $\hat{l}_t(\vec{\theta})$ and the corresponding cumulative loss be $\hat{L}_T(\vec{\theta})$. That is, $\hat{l}_t(\vec{\theta}) = l_t(\vec{\theta})$ if the employed policy is a theshold-based policy. It follows that

$$\hat{l}_t(\vec{\theta}) = \begin{cases} \beta_t & \text{if } \theta_l \le f_t < \theta_u, \\ \phi_t & \text{otherwise.} \end{cases} \tag{3}$$

Define $\vec{\theta}^* = (\theta_l^*, \theta_u^*)$ as the fixed thresholds with the lowest expected cumulative loss in the class of fixed-$\vec{\theta}$ policies.

$$\vec{\theta}^* = \arg\min_{\vec{\theta}} \mathbb{E}[\hat{L}_T(\vec{\theta})] = \arg\min_{\vec{\theta}} \mathbb{E}\left[\sum_{t=1}^{T} \hat{l}_t(\vec{\theta})\right]. \tag{4}$$

We use the fixed threshold policy with threshold $\vec{\theta}^*$ as the reference policy to compare the performance of candidate policies. Our performance metric for a candidate policy is regret, denoted by R_T, where

$$R_T = \mathbb{E}[L_T] - \mathbb{E}[\hat{L}_T(\vec{\theta}^*)]. \tag{5}$$

The expectation in (4) and (5) are taken over the randomness in the arrivals x_t and in the candidate policy.

3 Optimal Policy Under a Calibrated Model

In this section, we focus on the setting where the LDL is calibrated, i.e., for each sample, the LDL output (softmax value) for each class equals the a posteriori probability of that sample belonging to that class. Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input space and the label space, respectively. Assume that $(x_t, h_r(x_t))$ is drawn i.i.d from $\mathcal{X} \times \mathcal{Y}$.

Definition 1 (Calibrated Model)**.** *The DL model that generates the softmax values $(1 - f_t, f_t)$ for the sample x_t is calibrated, if $\mathbb{P}(h_r(x_t) = 1 | x_t) = f_t$, $\forall t$.*

We now characterise the optimal HI offloading policy for cost-sensitive classification for calibrated LDL.

Theorem 1. *Let x_t denote the sample input to a calibrated LDL. Then, the optimal predictor $h_l^*(x_t)$ is given by*

$$h_l^*(x_t) = \begin{cases} 1, & \text{if } f_t \geq \frac{\delta_1}{\delta_1+\delta_{-1}}, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Further, with a known β_t, the optimum offloading decision for this setup is to offload the sample when

$$\theta_l^{*(t)} := \frac{\beta_t}{\delta_{-1}} \leq f_t < 1 - \frac{\beta_t}{\delta_1} := \theta_u^{*(t)}, \tag{7}$$

with an associated expected loss of

$$\mathbb{E}[\hat{l}_t] = \min\{\beta_t, \delta_1(1-f_t), \delta_{-1} f_t\}. \tag{8}$$

Remark 1. *Theorem 1 implies the following: (i) Offloading does not occur if $\beta_t \geq \frac{\delta_1 \delta_{-1}}{\delta_1+\delta_{-1}}$, half the harmonic mean of δ_1 and δ_{-1}; (ii) when $\delta_1 = \delta_{-1}$, the decision is equivalent to Chow's rule for classification with rejection, with no offloading if $\beta_t \geq 0.5$, and offloading if and only if $\min\{f_1(x_t), f_0(x_t)\} > \beta_t$. If not offloaded, the prediction is the class with the higher softmax value.*

The key question is its extension to non-calibrated models, where the thresholds must be learned.

4 The H2T2 Policy

This section relaxes the assumption of a calibrated LDL, proposing instead an online policy that learns the values of the two thresholds. Any policy within our framework must make two decisions: first, whether to offload a given sample; and second, if the sample is not offloaded, the policy has to predict a class based on the LDL's output.

Algorithm We present the proposed HI-Hedge with Two Thresholds (H2T2) policy in Algorithm 1. We note that H2T2 maps to the Prediction with Expert Advice (PEA) problem, where the *experts* are the threshold tuples $\vec{\theta}$, and the expert chosen in round t and $\vec{\theta}^{(t)} := (\theta_l^{(t)}, \theta_u^{(t)})$. We denote the set of experts as Θ. Given $\theta_l \leq \theta_u$, it follows that $|\Theta| = 2^{b-1}(2^b+1)$ experts. Fig. 3 illustrates three regions of the experts with respect to an observed f_t. We maintain weights for all experts. The probability of choosing an expert in a round is proportional to its weight in that round. This process will be discussed shortly.

Under H2T2, given the output of the LDL and the expert (i.e., threshold tuple) chosen in round t, denoted with superscript $^{(t)}$, the local prediction $h_l(x_t)$ is defined as follows:

$$h_l(x_t) = \begin{cases} 0, & \text{if } f_t < \theta_l^{(t)} \leq \theta_u^{(t)}, \\ \text{Ambiguous}, & \text{if } \theta_l^{(t)} \leq f_t < \theta_u^{(t)}, \\ 1, & \text{if } \theta_l^{(t)} \leq \theta_u^{(t)} \leq f_t. \end{cases} \tag{9}$$

For simplicity, we drop the superscript $\cdot^{(t)}$ here onward, when the time index is already implied. H2T2 offloads all samples for which $h_l(x_t)$ is ambiguous. Otherwise, it still offloads with a probability ϵ based on a Bernoulli random variable $\zeta_t \sim \text{Ber}(\epsilon)$. The motivation of this *exploration* step will be explained shortly. If H2T2 does not offload, $h_l(x_t)$ is the predicted class.

Algorithm 1: H2T2: The two threshold HI policy.

ϵ, η, Θ. ▷ Input
$w_1(\vec{\theta}) = 1 \;\forall\, \vec{\theta} \in \Theta$, and $W_1 = |\Theta|$. ▷ Initialization
for $t \geq 1$ **do**
    Observe f_t. ▷ LDL inference
    $p_t = \sum_{\theta_l < f_t} \sum_{\theta_u < f_t} w_t(\vec{\theta})$. ▷ Region 3
    $q_t = \sum_{\theta_l \leq f_t} \sum_{\theta_u > f_t} w_t(\vec{\theta})$. ▷ Region 2
    $\psi_t \sim \text{Uniform}(0,1)$.
    $\zeta_t \sim \text{Bernoulli}(\epsilon)$.
    if $\psi_t \leq q_t$ or $\zeta_t = 1$ **then** ▷ Region 2, or exploration
        Offload x_t, receive $h_r(x_t)$, and observe ϕ_t.
    else ▷ Local inference
        if $\psi_t \leq q_t + p_t$ **then**
            $h_l(x_t) = 1$. ▷ Region 3
        else
            $h_l(x_t) = 0$. ▷ Region 1
        end if
    end if
    for all $\vec{\theta} \in \Theta$ **do**
        Compute $\tilde{l}_t(\vec{\theta})$ using (11).
        $w_{t+1}(\vec{\theta}) = e^{-\eta \tilde{l}_t(\vec{\theta})} w_t(\vec{\theta})$. ▷ Weight updation
    end for
    $W_{t+1} = \sum_{\vec{\theta} \in \Theta} w_{t+1}(\vec{\theta})$.
end for

We now discuss the process of computing the weights under H2T2. Recall that, in our setting, we use the inference of the RDL as a proxy for the ground truth. One of the key challenges in our setting is that this inference is only observed when we offload a sample; therefore, unlike the classical PEA setting, we learn the loss incurred by the experts only when we offload a sample.

To address this challenge, H2T2 employs two steps: First is the exploration step discussed earlier, where unambiguous samples are randomly offloaded with a probability ϵ. Next, H2T2 maintains an estimated loss $\tilde{l}_t(\vec{\theta})$ for each expert that is used for weight updation. Let $\mathbb{1}_{x_t}(\vec{\theta})$ be an indicator random variable that is one if $h_l(x_t)$ would have been ambiguous if the expert chosen in time t was $\vec{\theta}$, i.e., $\theta_l \leq f_t < \theta_u$.

First we write the true loss from (2), under H2T2:

$$l_t(\vec{\theta}) = \begin{cases} \phi_t & \text{if } \mathbb{1}_{x_t}(\vec{\theta}) = 0, \zeta_t = 0, \\ \beta_t & \text{otherwise.} \end{cases} \tag{10}$$

Next, we define the estimated loss:

$$\tilde{l}_t(\vec{\theta}) = \begin{cases} \beta_t & \text{if } \mathbb{1}_{x_t}(\vec{\theta}) = 1, \\ \frac{\phi_t}{\epsilon} & \text{if } \mathbb{1}_{x_t}(\vec{\theta}) = 0, \zeta_t = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Also, let $\tilde{L}_T = \sum_{t=1}^{T} \tilde{l}_t(\vec{\theta})$ be the corresponding cumulative estimated loss. Each expert's weight in a round is an exponential function of the cumulative estimated loss of that expert up to that round, with a learning rate η. Note that this estimated loss $\tilde{l}_t$ is used as part of the H2T2 policy for weight

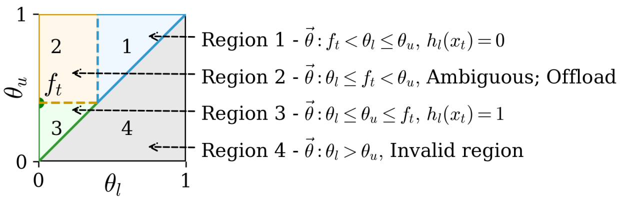


Figure 3: Illustration showing the invalid and three valid regions of experts from (9) with respect to an observed f_t.

updation and is different from the true loss l_t incurred by the algorithm. In our setting, all experts that yield the same $h_l(x_t)$ create contiguous and convex regions, as illustrated in Fig. 3. This simplifies the decision-making process: it suffices to compute the total probability of all experts in each of the three regions to decide whether to offload and to determine the inference of the LDL.

Regret analysis We now provide regret guarantees for H2T2. We first state the following two lemmas. Lemma 1 states that $\tilde{l}_t(\vec{\theta})$ is an unbiased estimator of the losses incurred by a threshold-based policy $\hat{l}_t(\vec{\theta})$ in (3), used for regret computation.

Lemma 1. *For all values of f_t and for all $\vec{\theta} \in \Theta$:*

$$\mathbb{E}_{\zeta}[\tilde{l}_t(\vec{\theta})] = \hat{l}_t(\vec{\theta}).$$

Next, in Lemma 2, we bound the difference in the expected values of the cumulative true loss (see (10)) and the cumulative loss of the baseline policy, where the expectation is over the randomness of the policy, that is over both the randomness of the expert selection and the exploration with ζ.

Lemma 2. *Under H2T2 and a time horizon T, for all $\vec{\theta} \in \Theta$:*

$$\mathbb{E}\left[L_T\right] - \mathbb{E}\left[\hat{L}_T(\vec{\theta})\right] \leq \epsilon\beta T \leq \epsilon T.$$

We use these lemmas to upper bound the regret of H2T2.

Theorem 2. *The regret R_T of H2T2 satisfies*

$$R_T \leq \left(\epsilon\beta + \tfrac{\eta}{2\epsilon}\right) T + \tfrac{\ln(|\Theta|)}{\eta}. \tag{12}$$

The regret-minimizing values of η and ϵ can be obtained by setting the partial derivatives of R_T to zero.

Corollary 1. *R_T is minimized by $\epsilon^* = \left(\ln(|\Theta|)/2\beta^2 T\right)^{1/3}$ and $\eta^* = \sqrt{2\epsilon^* \ln|\Theta|/T}$. H2T2 with η^* and ϵ^* has $O\left(T^{2/3}\right)$ regret.*

We thus note that H2T2 has sublinear regret.

5 Experimental Results

We evaluate the performance of H2T2 via experiments on a MacBook Pro (Apple M1, 16GB, macOS 12.0.1) using the following dataset-model pairs, which we refer to simply by the dataset name.

(a) The Breast Cancer Histopathological Image Classification **(BreakHis)** (Spanhol et al. 2015) contains 7909 images of benign (class 1) and malignant cells. The LDL model is a lightweight MobileNet-based classifier with additional dense and dropout layers.

(b) **Chest** (Mohamed 2025) contains 844 chest CT scan images re-categorised into healthy and cancerous (class 1) classes for our use. The LDL is based on MobileNet, with a size of 13.3 MB including trainable parameters.

(c) **Phishing** (Tiwari 2025; Tan 2018) includes data collected from 5000 phishing and as many legitimate websites, with additional samples for testing. A logistic regression model (56 bytes) is used as LDL.

(d) For the **Synthetic** dataset, we generate softmax-like values using Gaussian mixtures truncated to $(0, 1)$.

(e) **BreaCh** is created by classifying all 7909 BreakHis samples using the model trained on Chest CT scan. It mimics an OOD data scenario with a domain shift for the model trained on Chest CT scan. As a result, unbeknownst to the user, the accuracy drops below 50%, resulting in the misdiagnosis of 38% cancerous tumors.

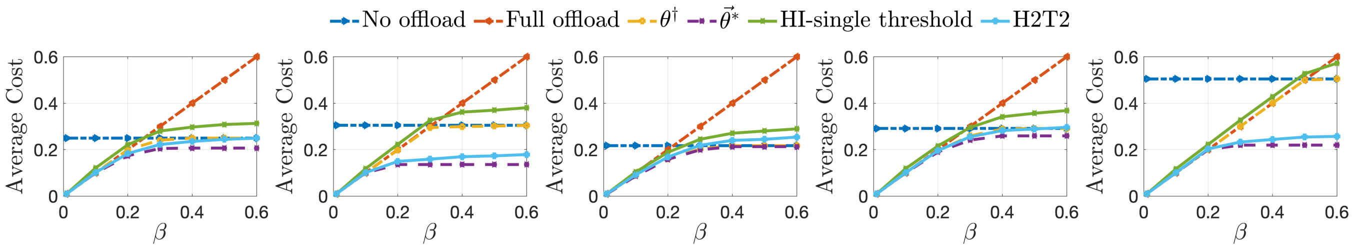
These are summarised in Table 2. These datasets and models are chosen to span diverse domains that have asymmetric misclassification costs and also show an OOD application. For benchmarking, we implement the following policies:

1. *No Offload*, where the LDL inference is accepted as is;
2. *Full Offload*, where all samples are offloaded;
3. *HI-single threshold*, the state-of-the-art continuum expert HI policy (Moothedath, Champati, and Gross 2024);
4. $\theta^{\dagger}$, offline optimal single-threshold policy;
5. $\vec{\theta}^{*}$, offline optimal two-threshold policy;
6. *H2T2*, our proposed policy.

We set $\delta_1 = 0.7$ and $\delta_{-1} = 1$ and use a fixed offloading cost β for comparison study. Similar trends are observed for varying δ_1 and δ_{-1}. The quantised threshold resolution is set to 0.0625, resulting in $|\Theta| < 256$. An increase in resolution did not result in a considerable reduction in average cost for the datasets studied. Since the bound optimising parameters from 1 do not necessarily minimise regret, we use η^* from Corollary 1 and set learning rate $\eta = 1$ without further parameter tuning. For fair comparison, $T = 10^4$ samples from each dataset via uniform sampling. Standard deviation for HI-single threshold and H2T2 are included in all figures, but are negligible (25 algorithm runs).

Dataset	Size	LDL	Accuracy	FP	FN
BreakHis	3365	MobileNet-based	72%	10%	18%
Chest	278	MobileNet-based	64%	16%	20%
Phishing	1106	Logistic regression	75%	12%	13%
Synthetic	10^5	Naive	66%	15%	19%
BreaCh	7909	Chest classifier	45%	17%	38%

Table 2: Different dataset-model pairs used in Fig. 4. Note that BreaCh is intentionally made erroneous.



(a) BreakHis (b) Chest (c) Phishing (d) Synthetic (e) BreaCh

Figure 4: Average cost of H2T2 policy vs. fixed offloading cost β for different datasets. Figures (a)-(d) are for in-distribution data, and (e) is for OOD data.

Performance Comparison

In Fig. 4, we present the average cost incurred by different policies by varying the fixed offloading cost $\beta \in (0, 0.6)$. We observe that H2T2 outperforms the state-of-the-art HI single-threshold policy and provides a percentage reduction of up to 55% (for Fig. 4(d) at $\beta = 0.6$) for the dataset, model, parameters, and range of β used in this illustration. However, for Phishing in Fig. 4(c), the improvement is marginal. This is attributed to the policies yielding near-trivial results, as indicated by the overlap of their respective offline optima and the No-offload policy. Interestingly, in two datasets, H2T2 outperforms $\theta^\dagger$, the offline optimal single-threshold policy.

For some β values, the static policies outperform online learning policies (e.g., No offload policy with $\beta > 0.3$ in Phishing). However, these regions are dataset-and model-dependent and unknown a priori.

Note the significant cost reduction achieved by H2T2 for BreaCh in Fig. 4(e) for the OOD data. Being model-agnostic and robust to distribution shifts, H2T2 limits the additional cost of data or model mismatches to about 0.08 when comparing Fig. 4(e) with 4(b), or to about 0.01 when comparing Fig. 4(e) with 4(a).

Observe the FNs and FPs of the LDL without the proposed H2T2 in Table 2. In our evaluation, we observed that H2T2 learns and adapts to reduce the costlier FNs even when the offload is relatively costly ($\beta = 0.6$). Specifically, for the BreakHis dataset, FNs were reduced to 9.5%, while False Positives (FPs) increased to 14%. In the OOD setting of BreaCh, H2T2 was particularly effective, reducing FNs to below 1%, with FPs rising to 28%.

6 Discussion: Multiclass Classification

We have thus far focused on cost-sensitive binary classification. We now briefly examine extending our setting to multiclass classification with asymmetric pairwise misclassification costs. The overarching objective remains the same: to partition the softmax value space into distinct regions for each class and an additional region for offload.

Let K be the number of classes and $C \in [0,1]^{K\times K}$ be the normalized cost matrix where C_{ij} is the cost of misclassifying class i as class j, and $C_{ii} = 0, \forall i$. Let f be the $K \times 1$ softmax vector and C_k denote the k^{th} column of C, representing the cost vector when class k is the prediction. The loss function l_t follows (2), with a modified ϕ_t defined as:

$$\phi_t = C_{ij}, \quad \text{if class } i \text{ is predicted as } j. \tag{13}$$

Theorem 3. *The optimal predictor $h_l^*(x_t)$ of a calibrated LDL designed for a K-class classification task is given by:*

$$h_l^*(x_t) = \arg\min_k f^T C_k. \tag{14}$$

Further, the optimum offloading decision for this setup is to offload the sample when $\min_k f^T C_k > \beta_t$ with an associated expected cost of $\mathbb{E}[l_t] = \min\{\beta_t, \min_k f^T C_k\}$.

For calibrated models, this yields $K+1$ decision regions bounded by $(K-2)$-dimensional simplexes in closed-form. For instance, in the binary case ($K = 2$) of previous sections, we saw three regions bounded by 2 points; for $K = 3$, there are four regions bounded by line segments as illustrated in Fig. 5. Note that we could represent these regions in K-dimensions on the $K-1$-dimensional standard (probability) simplex $f^T\vec{1} = 1$, or remove an independent variable and represent it directly in $K-1$ dimensions.

In the case of uncalibrated models, the boundaries of these regions are not known a priori. Similar to the binary setting, we consider a class of experts for each of these boundaries, and design a similar PEA policy to learn them. Designing a compact and scalable methodology that utilises the special structure of costs to determine the costs of all experts in a computationally efficient manner is an open problem.

7 Conclusion

We study cost-sensitive binary classification using a local model that works in conjunction with a more accurate but costly remote model. We use the HI framework, where each sample is first sent to the local model, and the decision to offload to the remote model is made based on the output of the local model. For calibrated local models, we derived a Bayes optimal two-threshold rule. For general, uncalibrated models, we developed H2T2, an online learning policy that finds the optimal thresholds with partial feedback and has sublinear regret. Unlike previous HI learning algorithms that use a single threshold for uniform misclassification costs, H2T2 employs two thresholds to handle asymmetric costs, thereby creating an ambiguity region for the offloading decision. It also makes its own cost-sensitive local inferences. H2T2 is a novel approach because it operates during the inference phase without requiring LDL retraining to improve

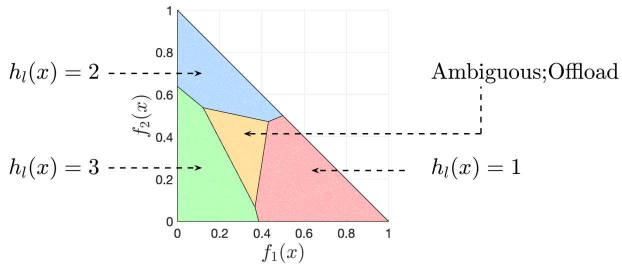


Figure 5: Illustration showing the four regions for a three-class classification by a calibrated LDL, plotted in a plane defined by the (first) two independent softmax components. Linear boundaries bound the regions, and the inferences are marked within. $\delta_1 = 0.7, \delta_{-1} = 1, \beta = 0.4$. A suitable cost matrix C is selected from a random sample for clarity.

the classification accuracy. It is also agnostic to the models being used. Simulations across various datasets and models show that H2T2 consistently outperforms naive, single-threshold, and even offline single-threshold policies. While certain dataset-model combinations might favour naive policies, these conditions are unknown beforehand and are vulnerable to OOD data. H2T2 is a robust and flexible solution for OOD data, making it ideal for real-world applications.

A Proofs

Proof of Theorem 1 on page 4: Define an optimal predictor as the one that minimises the expected LDL cost. Let Y denote random variable $h_r(X)$. Recall ϕ_t from (1) and $f_t \coloneqq f_1(x_t)$.

$$\begin{aligned}\mathbb{E}[\phi_t] &= \mathbb{E}_X\big[\delta_1 \mathbb{1}_{h_l(X)=1}\mathbb{P}(Y=0 \mid X) \\ &\quad + \delta_{-1}\mathbb{1}_{h_l(X)=0}\mathbb{P}(Y=1 \mid X)\big], \\ &= \mathbb{E}_X\big[\delta_1 \mathbb{1}_{h_l(X)=1}(1-f(X)) + \delta_{-1}\mathbb{1}_{h_l(X)=0} f(X)\big],\end{aligned}$$

where $\mathbb{1}_.$ is the indicator function. It follows that the optimal predictor of the calibrated LDL, denoted by $h_l^*(X)$, activates the indicator function that minimises $\mathbb{E}[\phi_t]$. That is,

$$h_l^*(X) = \begin{cases} 1, & \text{if} \quad \Rightarrow f(X) \geq \frac{\delta_1}{\delta_1+\delta_{-1}}, \\ 0, & \text{otherwise,}\end{cases}$$

as given in (6). Thus, with this optimum predictor, we get:

$$\mathbb{E}[\phi_t] = \min\{\delta_1(1-f_t), \delta_{-1}f_t\}.$$

Now, let $\mathbb{I}_t$ denote the indicator random variable denoting the decision to offload sample x_t. Then we have the cost l_t:

$$l_t = \beta_t \mathbb{I}_t + \phi_t(1-\mathbb{I}_t)$$

with an the optimum decision is to offload if $\beta_t < \mathbb{E}[\phi_t]$. That is, for calibrated LDL with an optimal predictor, the optimum decision is to offload when

$$\begin{aligned}\beta_t &< \min\{\delta_1(1-f_t), \delta_{-1}f_t\}, \\ \Rightarrow \theta_l^{*(t)} &= \tfrac{\beta_t}{\delta_{-1}} < f_t < 1 - \tfrac{\beta_t}{\delta_1} = \theta_u^{*(t)},\end{aligned}$$

with an associated expected cost given by (8). □

Proof of Lemma 1 on page 5: Taking expectation of $\hat{l}$ over realizations of ζ gives

$$\begin{aligned}\mathbb{E}_\zeta[\tilde{l}_t(\vec{\theta})] &= \mathbb{1}_{f_t \notin [\theta_l,\theta_u]} \cdot \mathbb{P}(\zeta=1)\tfrac{\phi_t}{\epsilon} + \mathbb{1}_{\theta_l < f_t < \theta_u} \cdot \beta_t, \\ &= \mathbb{1}_{f_t \notin [\theta_l,\theta_u]} \cdot \phi_t + \mathbb{1}_{\theta_l < f_t < \theta_u} \cdot \beta_t = \hat{l}_t(\vec{\theta}).\end{aligned}$$

That is, the estimated loss is an unbiased estimator of the losses incurred by a threshold-based policy. □

Proof of Lemma 2 on page 5: Consider a the algorithm choosing an expert in round t, with losses $l_t(\vec{\theta})$ and $\hat{l}_t(\vec{\theta})$. We have,

$$\mathbb{E}[L_T] = \sum_{t=1}^{T}\sum_{\vec{\theta}\in\Theta} \tfrac{w_t(\vec{\theta})}{W_t}\mathbb{E}_{\zeta_t}\left[l_t(\vec{\theta})\right]. \tag{15}$$

Now, from (3) and (10),

$$\begin{aligned}\hat{l}_t(\vec{\theta}) &= \mathbb{1}_{f_t \notin [\theta_l,\theta_u]}\phi_t + \left(1 - \mathbb{1}_{f_t \notin [\theta_l,\theta_u]}\right)\beta_t, \\ l_t(\vec{\theta}) &= \mathbb{1}_{f_t \notin [\theta_l,\theta_u]}\mathbb{1}_{\zeta_t=0}\phi_t + \left(1 - \mathbb{1}_{f_t \notin [\theta_l,\theta_u]}\mathbb{1}_{\zeta_t=0}\right)\beta_t.\end{aligned}$$

Taking expectation and combining, we get,

$$\mathbb{E}_\zeta\left[l_t(\vec{\theta})\right] - \mathbb{E}_\zeta\left[\hat{l}_t(\vec{\theta})\right] = \epsilon\mathbb{1}_{f_t \notin [\theta_l,\theta_u]}(\beta_t - \phi_t) \leq \epsilon\beta_t.$$

In the last step, we used Lemma 1. Since this holds for any $\vec{\theta}$ and t, we have for any policy and time horizon T

$$\mathbb{E}[L_T] - \mathbb{E}\left[\hat{L}_T(\vec{\theta})\right] \leq \epsilon\beta_t T \leq \epsilon\beta T \leq \epsilon T. \qquad \square$$

Proof of Theorem 2 on page 5: Recall $\tilde{L}_T(\vec{\theta})$: the cumulative estimated loss with a fixed $\vec{\theta}$. Consider $\ln\frac{W_{T+1}}{W_1}$:

$$\begin{aligned}\ln\tfrac{W_{T+1}}{W_1} &= \ln\sum_{\vec{\theta}\in\Theta} e^{-\eta\sum_{t=1}^{T}\tilde{l}_t(\vec{\theta})} - \ln(|\Theta|), \\ &\geq -\min_{\vec{\theta}\in\Theta}\eta\tilde{L}_T(\vec{\theta}) - \ln(|\Theta|).\end{aligned} \tag{16}$$

Using Lemma 1 and Jensen's inequality on expectations:

$$\begin{aligned}\mathbb{E}_\zeta\left[\ln\tfrac{W_{T+1}}{W_1}\right] &\geq -\min_{\vec{\theta}\in\Theta}\eta\mathbb{E}_\zeta\left[\tilde{L}_T(\vec{\theta})\right] - \ln(|\Theta|) \\ &= -\min_{\vec{\theta}\in\Theta}\eta\hat{L}_T(\vec{\theta}) - \ln(|\Theta|).\end{aligned} \tag{17}$$

Next, we upper bound $\log\frac{W_{T+1}}{W_1}$ as follows:

$$\begin{aligned}\ln\tfrac{W_{t+1}}{W_t} &= \ln\sum_{\vec{\theta}\in\Theta}\tfrac{w_t(\vec{\theta})}{W_t}e^{-\eta\tilde{l}_t(\vec{\theta})} \\ &\leq \ln\sum_{\vec{\theta}\in\Theta}\tfrac{w_t(\vec{\theta})}{W_t}\left(1 - \eta\tilde{l}_t(\vec{\theta}) + \left(\eta\tilde{l}_t(\vec{\theta})\right)^2/2\right) \qquad (18) \\ &= \ln\left(1 + \sum_{\vec{\theta}\in\Theta}\tfrac{w_t(\vec{\theta})}{W_t}\left(-\eta\tilde{l}_t(\vec{\theta}) + \left(\eta\tilde{l}_t(\vec{\theta})\right)^2/2\right)\right).\end{aligned}$$

Simplifying, we get,

$$\begin{aligned}\ln\tfrac{W_{t+1}}{W_t} &\leq \sum_{\vec{\theta}\in\Theta}\tfrac{w_t(\vec{\theta})}{W_t}\left(-\eta\tilde{l}_t(\vec{\theta}) + \left(\eta\tilde{l}_t(\vec{\theta})\right)^2/2\right) \qquad (19) \\ &\leq \sum_{\vec{\theta}\in\Theta}\tfrac{w_t(\vec{\theta})}{W_t}\left(-\eta\tilde{l}_t(\vec{\theta}) + \eta^2\tilde{l}_t(\vec{\theta})/2\epsilon\right). \qquad (20)\end{aligned}$$

Above, we have used $e^{-x} \leq 1 - x + \frac{x^2}{2}$, $\ln(1+x) \leq x$, and $\tilde{l}_t(\vec{\theta}) \leq \frac{1}{\epsilon}$ in (18), (19), and (20), respectively. Now we sum (20) telescopically over the entire horizon T as follows:

$$\ln \frac{W_{T+1}}{W_1} = \ln \prod_{t=1}^{T} \frac{W_{t+1}}{W_t} = \sum_{t=1}^{T} \ln \frac{W_{t+1}}{W_t}, \quad (21)$$

$$\leq \sum_{t=1}^{T} \sum_{\vec{\theta} \in \Theta} \frac{w_t(\vec{\theta})}{W_t} \left(-\eta \tilde{l}_t(\vec{\theta}) + \frac{\eta^2 \tilde{l}_t(\vec{\theta})}{2\epsilon} \right). \quad (22)$$

$$\Rightarrow \mathbb{E}\Big[\ln \frac{W_{T+1}}{W_1}\Big] \leq -\eta \sum_{t=1}^{T} \sum_{\vec{\theta} \in \Theta} \frac{w_t(\vec{\theta})}{W_t} \mathbb{E}_\zeta[\tilde{l}_t(\vec{\theta})]$$

$$+ \eta^2 \sum_{t=1}^{T} \sum_{\vec{\theta} \in \Theta} \frac{w_t(\vec{\theta})}{W_t} \frac{\mathbb{E}_\zeta[\tilde{l}_t(\vec{\theta})]}{2\epsilon}$$

$$\leq -\eta \mathbb{E}[\hat{L}_T(\vec{\theta})] + \frac{\eta^2}{2\epsilon} T \quad (23)$$

$$\leq -\eta \mathbb{E}[L_T(\vec{\theta})] + \eta \epsilon \beta T + \frac{\eta^2}{2\epsilon} T. \quad (24)$$

In (23), we use $\mathbb{E}_\zeta[\tilde{l}_t(\vec{\theta})] = \hat{l}_t(\vec{\theta}) \leq 1$ and the expectation is taken over expert selection and ζ. In (24), we use Lemma 2. The result follows by comparing bounds from (17) and (24):

$$\eta \mathbb{E}[L_T(\vec{\theta})] - \min_{\vec{\theta} \in \Theta} \eta \hat{L}_T(\vec{\theta}) \leq \eta \epsilon \beta T + \frac{\eta^2 T}{2\epsilon} + \log(|\Theta|).$$

Recall (5). Taking expectation over arrivals $\mathcal{X}$, we get:

$$R_T \leq \left(\epsilon \beta + \frac{\eta}{2\epsilon}\right) T + \frac{\log(|\Theta|)}{\eta}.$$

Corollary 1 follows by setting partial derivatives to zero. While not optimal, choosing a simpler $\epsilon = \sqrt{\eta/2}$ and $\eta = \left(2 \ln^2(|\Theta|)/T^2\right)^{1/3}$ also yields regret in the same order: $O(T^{2/3})$. □

Remark: While our aim was to bound the expected regret over i.i.d. arrivals, a stronger minimax result for any input sequence can be obtained by omitting the expectation over $\mathcal{X}$ in the final step.

Proof of Theorem 3 on page 6: Let X and Y denote random variables of x_t and $h_r(x_t)$, respectively.

$$\mathbb{E}[\phi_t] = \mathbb{E}_X \sum_{j=1}^{K} \mathbb{1}_{h_l(X)=j} \sum_{i=1}^{K} \mathbb{P}(Y = i \mid X) C_{ij},$$

$$= \mathbb{E}_X \sum_{j=1}^{K} \mathbb{1}_{h_l(X)=j} f^T C_j.$$

It follows that $h_l^*(X)$ activates the j that minimizes $f^T C_j$:

$$h_l^*(X) = \arg\min_j f^T C_j,$$

with $\mathbb{E}[\phi_t] = \min_j f^T C_j$. Now, let $\mathbb{I}_t$ denote the decision to offload x_t. We have, $l_t = \beta_t \mathbb{I}_t + \phi_t(1 - \mathbb{I}_t)$. Thus, it is optimal to offload if $\beta_t < \mathbb{E}[\phi_t]$. That is, for calibrated LDL with optimal predictor, it is optimal offload when $\beta_t < \min_k \{f^T C_k\}$, with $\mathbb{E}[l_t] = \min\{\beta_t, \min_k f^T C_k\}$. □

Acknowledgements

This work is supported in part by the Vetenskap Radet (VR)-Optimal Sampling for Interactive Networked Applications under Grant 2022-03922, in part by the NordForsk Nordic University Cooperation on Edge Intelligence under grant number 168043, in part by Natural Sciences and Engineering Research Council (NSERC) of Canada Discovery Grant No. 54196-52150, and in part by an SCPP grant from IIT Bombay, India.

References

Al-Atat, G.; Datta, P.; Moharir, S.; and Champati, J. P. 2024. Regret bounds for online learning for hierarchical inference. In *Proceedings of the 25th MobiHoc '24*, 281–290. ACM.

Al-Atat, G.; Fresa, A.; Behera, A. P.; Moothedath, V. N.; Gross, J.; and Champati, J. P. 2023. The Case for Hierarchical Deep Learning Inference at the Network Edge. In *Proceedings of the 1st NetAISys*. ACM.

Behera, A. P.; Daubaris, P.; Bravo, I.; Gallego, J.; Morabito, R.; Widmer, J.; and Champati, J. P. 2025. Exploring the Boundaries of On-Device Inference: When Tiny Falls Short, Go Hierarchical. *IEEE Internet of Things Journal*, 1–1.

Beytur, H. B.; Aydin, A. G.; de Veciana, G.; and Vikalo, H. 2024. Optimization of Offloading Policies for Accuracy-Delay Tradeoffs in Hierarchical Inference. In *IEEE INFOCOM 2024*, 1989–1998.

Cesa-Bianchi, N.; and Lugosi, G. 2006. *Prediction, learning, and games.* Cambridge University Press. ISBN 978-0-511-54692-1.

Chow, C. 2003. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16(1): 41–46.

Cortes, C.; DeSalvo, G.; and Mohri, M. 2016. Learning with rejection. In *27th International conference, ALT*, 67–82. Springer.

Datta, P.; Moharir, S.; and Champati, J. P. 2025. Online Learning with Stochastically Partitioning Experts. In *Proc. UAI.*

Elkan, C. 2001. The foundations of cost-sensitive learning. In *IJCAI*, 973–978. Lawrence Erlbaum Associates Ltd.

Letsiou, A.; Moothedath, V. N.; Behera, A. P.; Champati, J. P.; and Gross, J. 2024. Hierarchical Inference at the Edge: A Batch Processing Approach. In *2024 IEEE/ACM SEC*, 476–482.

Mohamed, H. 2025. Chest CT-Scan images Dataset.

Moothedath, V. N.; Champati, J. P.; and Gross, J. 2024. Getting the Best Out of Both Worlds: Algorithms for Hierarchical Inference at the Edge. *IEEE Transactions on Machine Learning in Communications and Networking*, 2: 280–297.

Narasimhan, H.; Jitkrittum, W.; Menon, A. K.; Rawat, A.; and Kumar, S. 2022. Post-hoc estimators for learning to defer to an expert. *Advances in Neural Information Processing Systems*, 35: 29292–29304.

Ni, C.; Charoenphakdee, N.; Honda, J.; and Sugiyama, M. 2019. On the calibration of multiclass classification with rejection. *Advances in neural information processing systems*, 32.

Nikoloska, I.; and Zlatanov, N. 2021. Data Selection Scheme for Energy Efficient Supervised Learning at IoT Nodes. *IEEE Comm. Let.*, 25(3): 859–863.

Ramaswamy, H. G.; Tewari, A.; and Agarwal, S. 2018. Consistent algorithms for multiclass classification with an abstain option. *Electronic Journal of Statistics*, 12(1): 530 – 554.

Saerens, M.; Latinne, P.; and Decaestecker, C. 2002. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1): 21–41.

Spanhol, F. A.; Oliveira, L. S.; Petitjean, C.; and Heutte, L. 2015. A dataset for breast cancer histopathological image classification. *Ieee transactions on biomedical engineering*, 63(7): 1455–1462.

Sturzinger, E.; and Satyanarayanan, M. 2024. Beyond Federated Learning: Survival-Critical Machine Learning. In *2024 IEEE/ACM SEC*, 483–489. IEEE.

Tan, C. L. 2018. Phishing dataset for machine learning: Feature evaluation. *Mendeley Data*, 1(8).

Tiwari, S. 2025. Phishing Dataset for Machine Learning.