

Feature-Sharing in Cascade Detection Systems with Multiple Applications

Long N. Le, *Student Member, IEEE*, and Douglas L. Jones, *Fellow, IEEE*

Abstract—Traditional distributed detection systems are often designed for a single target application. However, with the emergence of the Internet of Things (IoT) paradigm, next-generation systems are expected to be a shared infrastructure for multiple applications. To this end, we propose a modular, cascade design for resource-efficient, multi-task detection systems. Two (classes of) applications are considered in the system, a primary and a secondary one. The primary application has universal features that can be shared with other applications, to reduce the overall feature extraction cost, while the secondary application does not. In this setting, the two applications can collaborate via feature sharing. We provide a method to optimize the operation of the multi-application cascade system based on an accurate resource consumption model. In addition, the inherent uncertainties in feature models are articulated and taken into account. For evaluation, the twin-comparison argument is invoked, and it is shown that, with the optimal feature sharing strategy, a system can achieve $9\times$ resource saving and $1.43\times$ improvement in detection performance.

Index Terms—Feature sharing, cascade detection system, multiple applications, resource-aware optimization, Internet of Things.

I. INTRODUCTION

Traditional distributed detection systems are often designed for a single target application. However, with the emergence of the Internet of Things (IoT) paradigm, next-generation systems are expected to be a shared infrastructure for multiple applications, and hence require rethinking of the overall system design.

To support multiple tasks seamlessly, a detection system needs to be modularly designed, i.e. made up of components that are reusable for various applications with different objectives and constraints. A similar view is shared by the TerraSwarm Research Center [1], whose aim is to create software components that serve as building blocks for IoT application developers. Likewise, Atzori et al. [2] proposed a service-oriented architecture for the IoT, where an ecosystem of services lays the foundation for IoT applications to be built on top.

Like many system design problems, resource-efficiency is an important objective in the design of multi-task detection systems [3]. A well-known resource-efficient, modular design is the cascade structure, which consists of a collection of detection modules in tandem. The design has been applied successfully in the single-application context, e.g. face detection [4]. However, we propose that the cascade also has a great

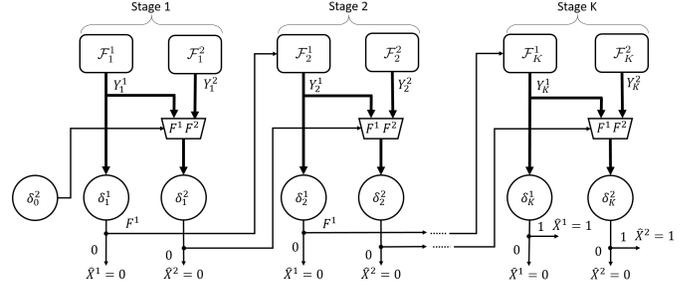


Fig. 1: The cascade detection system with 2 applications (indexed by superscripts) and K stages/layers (indexed by subscripts). For stage i of application j , \mathcal{F}_i^j denotes the feature extractor and δ_i^j denotes the binary decision of a detector. The feature itself is denoted by Y_i^j . X^j is the (detection) target state, and \hat{X}^j is the prediction about X^j by a detector.

potential in the multiple-application context. Namely, thanks to its modular design, modules in the cascade could be shared between applications. In addition, the output of a module can be used to dynamically guide/control the execution of others in the system, providing necessary degrees of freedom to optimize the trade-off between system resource consumption and detection performance.

In this paper, we specifically study the multi-application cascade detection system whose model is shown in Fig. 1. It is assumed that there are two applications and K layers in the system. The system is illustrated in Fig. 1, where the superscripts are used to index applications, and the subscripts are used to index stages.

Each layer of the cascade is occupied by detection modules/detectors for both applications. Ignoring the application index (superscript) for now, a detector at stage i consists of a feature extractor \mathcal{F}_i , which produces the feature Y_i , and a decision rule δ_i , which takes Y_i and all previous features Y_1, \dots, Y_{i-1} as input. δ_i outputs different values depending on both the application and the stage (see Eq. (1) and (2)). X is the state of the (detection) target, which takes value 1 when the target is present, and 0 otherwise. Finally, \hat{X} denotes the prediction of X by the detector.

Of the two applications, we let one be the *primary* and the other be the *secondary*, denoted by superscript indices 1 and 2 in Fig. 1, respectively. The primary application is the one whose feature extractors produce *universal features* that are sharable. In contrast, features produced by the secondary application are not universal and hence assumed to produce no value in sharing. This distinction is illustrated in Fig. 1, where

L.N. Le and D.L. Jones are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Illinois, IL, 61801 USA. D.L. Jones is currently Director of the Advanced Digital Sciences Center.

primary features $Y_i^1, i = 1, \dots, K$ can be shared with the secondary application, but secondary features $Y_i^2, i = 1, \dots, K$ can only be used by the secondary application. Examples of universal features for audio applications are signal energy, the Mel-frequency cepstrum coefficient (MFCC) [5] and time-varying sinusoidal features [6], which have been used extensively in various audio/speech inference applications. Examples of secondary features are internal representation in neural networks, such as that of autoencoders. It is worth noting that the definition of primary and secondary applications here has no relevance to the practical importance of each, and the two-application assumption is meant for simplicity, without loss of generality. In fact, our result can be easily generalized to an arbitrary number of applications in each class (primary,secondary).

The decisions of the primary application δ_i^1 can take on the following values depending on the layer/stage.

$$\delta_i^1 = \begin{cases} 0 : \text{stop and declare } \hat{X}^1 = 0 \text{ (negative)} \\ F^1 : \text{use the primary feature next} \end{cases} \quad (1)$$

$$i = 1, \dots, K - 1$$

$$\delta_K^1 = \begin{cases} 0 : \text{declare } \hat{X}^1 = 0 \text{ (negative)} \\ 1 : \text{declare } \hat{X}^1 = 1 \text{ (positive)} \end{cases}$$

Note that only negative decisions, i.e. $\hat{X} = 0$, are allowed at intermediate stages ($i = 1, \dots, K - 1$) since the goal is not to make the final decision (which is reserved for the last stage with the best performance) but to screen out early negative instances, which is more likely in a rare-target setting.

Since the secondary application has access to both primary and secondary features, its decisions δ_i^2 at intermediate stages has more options and are given as follows.

$$\delta_0^2 = \begin{cases} F^1 : \text{use the primary feature next} \\ F^2 : \text{use the secondary feature next} \end{cases}$$

$$\delta_i^2 = \begin{cases} 0 : \text{stop and declare } \hat{X}^2 = 0 \text{ (negative)} \\ F^1 : \text{use the primary feature next} \\ F^2 : \text{use the secondary feature next} \end{cases} \quad (2)$$

$$i = 1, \dots, K - 1$$

$$\delta_K^2 = \begin{cases} 0 : \text{declare } \hat{X}^2 = 0 \text{ (negative)} \\ 1 : \text{declare } \hat{X}^2 = 1 \text{ (positive)} \end{cases}$$

Namely, intermediate decisions include feature selection and (early) negative decision making. Note that δ_0^2 is the decision occurs before any features are observed, and thus restricted from making early negative decisions.

The cascade structure has been studied before in the literature. For instance, the seminal work by Viola and Jones [4] showed empirically that such a design is very effective in detecting rare targets in a large dataset, and was also proposed as a resource-efficient approach for stream mining by Turaga et al. in [7]. However, existing works either 1) offer solutions that have limitations, to be articulated in Section II, or 2) focus on a single application. Our study here involves the cascade structure with multiple applications, investigates the potential of sharing features between them, and offers a

solution that does not have limitations of prior works. For instance, our resource-consumption model is more accurate than existing works, which often suffers from a ‘nebulous’ resource-consumption model that is inapplicable in practice. Furthermore, it is observed that there are inherent uncertainties in some features of the cascade, and an approach is proposed to address them. Beside optimizing parameters of the cascade, we also show that, under mild conditions, the cascade design itself is optimal. That is, adding additional degrees of freedom such as early positive decisions to the cascade structure does not improve its performance.

The rest of the paper is organized as follows. Section II reviews prior works that studied the cascade structure, along with their limitations. Section III-A discusses feature models and their potential uncertainties, then Section III-B presents our formulation and solution for the multi-application cascade system. A system simulation and final remarks are given in Sections IV and V, respectively.

II. PRIOR WORKS

It is worthwhile to note that the cascade detection system of interest here is different from the serial detector network in the distributed detection literature [8]–[10], in which the decision of a current module is treated as an extra observation, instead of as a control signal to *sensor* subsequent modules and conserve resources.

The cascade architecture is prevalent in many inference applications, with the most widely-known example being the seminal work in face detection by Viola and Jones [4]. In [4], the system of cascaded detection modules is used to quickly discard many negative sub-images typically observed in face-detection applications, thus significantly speeding up the detection process. However, the cascade is not optimized in [4], leaving the optimal classifiers’ parameters, both thresholds and weights, to be desired.

To this end, Luo [11] proposed to optimize thresholds of each detection module in a cascade using the classical Neyman-Pearson criterion, without consideration of resource cost. Under the assumption of statistical independence between detection modules, a gradient-based algorithm is proposed to search for the locally optimal thresholds, which is also a limitation of [11]. In contrast, our approach guarantees a globally optimal, resource-aware solution and does not assume independence between stages.

Later, Jun and Jones [12] incorporated an energy resource constraint in the Neyman-Pearson-based optimization over thresholds of a two-stage cascade. In this setting, three solution types were identified: one that utilizes all of the available energy and false-alarm rate, one that utilizes all the energy while slacking the false-alarm constraint, and one that utilizes all the false alarm while slacking the energy constraint. An algorithm to find the optimal thresholds is only available if the true solution is of the first type. Later, it is proven in [13] that, if observations of the first stage are reused in the second stage, then the first-type solution is optimal. In contrast, our approach generalizes to an arbitrary number of stages.

Chen et al. [14] designed a surveillance system using a two-stage cascade of low-end (acoustic and infrared) and

high-quality (camera) sensors. The system in [14] can find a triggering threshold that either minimizes the detection error, or satisfies a constraint on the CPU utilization for video processing, but not both, and a heuristic was used to combine the two solutions, i.e. use the threshold that minimizes the detection error if it also satisfies the utilization constraint, otherwise use the one that satisfies the constraint. Unlike the ad-hoc approach of [14], our solution is derived from a well-defined framework. It is worth noting that Cohen et al. [15] also studied a similar problem in which a multi-modal sensing system (with a PIR sensor and a camera) was designed for monitoring vehicles. While the treatment in [15] is principled (based on the partially-observable Markov decision process (POMDP) framework), the sensors are *not* operated in cascade, but instead are equally plausible options at each time step, and hence is different from our work.

Since the optimization of the cascade is hard, Raykar et al. [16] relaxed the problem by assuming classifiers in the cascade produce soft/probabilistic outputs instead of hard decisions, and converted the joint optimization of classifiers' linear weights into a maximum *a posteriori* problem. Feature costs are also incorporated into the optimization using the standard Lagrangian argument, and a gradient-based algorithm is used to find the optimal weights. However, the thresholds must be found using an exhaustive grid search, which is computationally intensive for cascades with many classifiers. Our solution does not suffer this drawback.

Chen et al. [17] proposed a cyclic optimization algorithm to optimize the linear weights of the classifiers in the cascade, along with their early-exit thresholds. That is, at each iteration, the algorithm cycles through all classifiers in the cascade, optimizing each one while leaving others untouched. The algorithm stops when the loss function no longer improves. A disadvantage of such optimization procedure is that it requires multiple passes through the cascade, and there is no theoretical bound on the number of iterations it will take. In contrast, our solution requires only a single pass through the cascade.

In stream mining, Turaga et al. [7] employed a cascade of Gaussian mixture model (GMM)-based classifiers and formulated a problem to find both the number of mixture components and the threshold in each classifier that maximize the system detection rate subject to constraints on false alarm, memory and CPU. The solution in [7] takes a person-by-person approach where it iterates between 1) finding optimal numbers of mixture components, i.e. resource allocation, for all classifiers given thresholds, and 2) finding optimal thresholds for a given resource allocation. However, this approach failed to capture the direct dependence of the cascade's resource consumption on its thresholds, and is inherently suboptimal.

A limitation of the above works is that they only considered open-loop solutions where the thresholds are independent variables to be optimized. Ertin [18] considered closed-loop solutions for the two-stage cascade detection problem where the optimal decision rule at each stage, which is observation-dependent, is sought. It was shown that the optimal policies are still likelihood ratio tests, but with coupling thresholds, i.e. the threshold at a stage depends on the receiver operating characteristic (ROC) and the threshold of the other stage.

Namely, the optimal thresholds can not be found using the solution technique employed by [18]. Note that, unlike classical detection problems, optimizing thresholds in a cascade is critical in the trade-off between inference performance and resource cost. A contribution of this paper is finding the optimal parameters (both test-statistics and thresholds) for general cascade detection systems.

Trapeznikov et al. studied a generalization of the cascade that was termed multi-stage sequential reject classifier (MSRC), which is simply the cascade with an additional positive decision [19] or multiple additional (classification) decisions [20] at intermediate stages. Their resource-consumption model is 'nebulous', i.e. if the decision at an intermediate stage is to defer to the next stage, an abstract, *independent* "penalty" is incurred. In contrast, in our resource model, these penalties are shown to be precisely the Lagrangian-weighted of the feature extraction costs, and hence they are coupled (see Eq. (31)).

On the other hand, a resource-consumption model closely relates to ours was considered by Wang et al. in [21]. The minor difference is that, instead of being proposed, our model was derived from first principle. However, [21] formulated the problem using the empirical risk minimization framework since it was assumed that probabilistic models of high-dimensional features cannot be estimated. We take a different approach where it is assumed that probabilistic models of features *can* be estimated, by first reducing the features' dimensionality. In other words, the input into our algorithm are (probabilistic) models, not a dataset as in [21]. In addition, the solution proposed in [21] is a convex linear-program, which requires a convex relaxation (with an upper-bounding convex surrogate function) of the true objective function. In contrast, our solution is a dynamic program and requires no relaxation.

III. OPTIMIZING THE MULTIPLE-APPLICATION CASCADE SYSTEM

A. Feature models

The discussion in this section is applicable to both applications, and hence the application indices (superscripts) are dropped. For the rest of the document, the colon notation is used to denote a collection, e.g.

$$y_{1:i} \triangleq \{y_1, \dots, y_{i-1}, y_i\} \quad (3)$$

Recall that Y_i denotes the feature used by the detector at stage i , and is modeled as a random variable whose distribution depends on the latent target $X \in \{0, 1\}$, i.e.

$$Y_i \sim p_i(y_i|x), x \in \{0, 1\}, i = 1, \dots, K \quad (4)$$

where lower-case letters denote realizations of the corresponding random variable in upper case and p denotes a probability mass/density function. It is assumed that these distributions are stationary and hence can be estimated during training. To handle the non-stationarity case, a straightforward, yet naive method, is to perform periodic retraining. More sophisticated methods can be investigated in future work.

Using Bayes' rule, the posterior probability of target presence is given by

$$\begin{aligned}\pi_1(y_1) &= \frac{1}{1 + \frac{1-\pi_0}{l_1(y_1)\pi_0}} \\ \pi_i(y_{1:i}) &= \frac{1}{1 + \frac{1-\pi_{i-1}(y_{1:i-1})}{l_i(y_i)\pi_{i-1}(y_{1:i-1})}} \\ & \quad i = 2, \dots, K\end{aligned}\quad (5)$$

where $l_i(y_i) \triangleq p_i(y_i|1)/p_i(y_i|0)$ and $\pi_i(y_{1:i}) \triangleq P(X = 1|y_{1:i})$ are the likelihood function and posterior probability at stage i , respectively. $\pi_0 \triangleq P(X = 1)$ is the prior probability of the target presence. Finally, the evidence probability is given by

$$p_i(y_i|y_{1:i-1}) = p_i(y_i|1)\pi_{i-1} + p_i(y_i|0)(1 - \pi_{i-1}) \quad (6)$$

An important aspect of the cascade detection system is that, except for the last stage, the main goal of other stages is to quickly screen out negative instances, and not to make the final decision. Therefore features used at stages other than the last one are suboptimal for the detection task by design, to keep the cost of their execution low. For instance, the all-band energy feature can neither characterize a bandpass target precisely, nor distinguish between a bandpass target and another bandpass interference, but can still be useful in the cascade thanks to its low cost [22]. The sub-optimality of these early-stage features, either due to 1) the failure to discriminate the target against potential interferences, or 2) the insufficient modeling of the target, can all be modeled as *uncertainty* in feature models. To this end, we employ the following *least-favorable* feature density models, developed by Huber in the context of robust detection [23], [24, Chapter 10], [25, Chapter 6], in place of the nominal ones.

$$\begin{aligned}p_i(y|0) &\leftarrow \begin{cases} \frac{1-\epsilon_{0i}}{v'+w'l_{Li}} [v'p_i(y|0) + w'p_i(y|1)], l_{Li}(y) < l_{Li} \\ (1-\epsilon_{0i})p_i(y|0), l_{Li} \leq l_i(y) \leq l_{Ui} \\ \frac{1-\epsilon_{0i}}{w''+v''l_{Ui}} [w''p_i(y|0) + v''p_i(y|1)], l_i(y) > l_{Ui} \end{cases} \\ p_i(y|1) &\leftarrow \begin{cases} \frac{(1-\epsilon_{1i})l_{Li}}{v'+w'l_{Li}} [v'p_i(y|0) + w'p_i(y|1)], l_i(y) < l_{Li} \\ (1-\epsilon_{1i})p_i(y|1), l_{Li} \leq l_i(y) \leq l_{Ui} \\ \frac{(1-\epsilon_{1i})l_{Ui}}{w''+v''l_{Ui}} [w''p_i(y|0) + v''p_i(y|1)], l_i(y) > l_{Ui} \end{cases} \\ & \quad i = 1, \dots, K-1\end{aligned}\quad (7)$$

where the ' \leftarrow ' symbol is the assignment operator and

$$\begin{aligned}v' &= \frac{\epsilon_{1i} + \nu_{1i}}{1 - \epsilon_{1i}}, v'' = \frac{\epsilon_{0i} + \nu_{0i}}{1 - \epsilon_{0i}} \\ w' &= \frac{\nu_{0i}}{1 - \epsilon_{0i}}, w'' = \frac{\nu_{1i}}{1 - \epsilon_{1i}}\end{aligned}\quad (8)$$

and $0 \leq \epsilon_{0i}, \epsilon_{1i}, \nu_{0i}, \nu_{1i} \leq 1$ are uncertainty parameters of stage i . l_{Li} and l_{Ui} are the lower and upper bounds of the likelihood ratio at stage i , respectively, and can be found by solving the equations outlined in [25, Chapter 6]. Note that since the new least-favorable densities result in a bounded likelihood function, the corresponding posterior probability is also bounded.

$$\pi_{Li} \triangleq \frac{1}{1 + \frac{1-\pi_{i-1}}{l_{Li}\pi_{i-1}}} \leq \pi_i(y_{1:i}) \leq \pi_{Ui} \triangleq \frac{1}{1 + \frac{1-\pi_{i-1}}{l_{Ui}\pi_{i-1}}}\quad (9)$$

B. System model and optimization

Optimizing the cascade system amounts to finding optimal decision rules $\delta_{1:K}^{1:2}$ that jointly minimize the proposed system's Bayes risk R_B of incorrect decisions subject to an expected system resource (e.g. energy) constraint e .

$$\begin{aligned}\min_{\delta_{1:K}^{1:2}} R_B(\delta_{1:K}^{1:2}) &\triangleq \sum_{j=1}^2 R_B^j(\delta_{1:K}^j) \\ \text{s.t. } E(\delta_{1:K}^{1:2}) &\triangleq \sum_{j=1}^2 E^j(\delta_{1:K}^j) \leq e\end{aligned}\quad (10)$$

where E is the expected system resource consumption. It is assumed that the total Bayes risk and expected resource consumption of the system is the sum from those of the individual application, i.e. R_B^j and E^j , $j = 1, 2$. The Lagrangian technique can be used to convert the constrained optimization problem (10) into the following unconstrained, but regularized, one

$$\begin{aligned}\min_{\delta_{1:K}^{1:2}} R(\delta_{1:K}^{1:2}) &\triangleq \sum_{j=1}^2 R^j(\delta_{1:K}^j) \\ &\triangleq \sum_{j=1}^2 (\lambda E^j + R_{K,A}^j + \sum_{i=1}^K R_{i,M}^j)\end{aligned}\quad (11)$$

where the parameter λ , which depends on the resource constraint e , couples the resource consumptions of all stages together and R denotes the *system risk* (with R^j denotes the risk of application j), which is a measure of the combined detection performance and resource consumption. The Bayes risk R_B^j has been broken down into multiple terms. For application j , $R_{i,M}^j, i = 1, \dots, K-1$ are the miss (false negative) risks due to early negative decisions at intermediate stages. $R_{K,M}^j, R_{K,A}^j$ are the miss and false-alarm (false positive) risks due to incorrect decisions at the last stage. Note that the system has no false-alarm risk at intermediate stages, since the cascade structure does not allow early positive decisions to be made. Such a constraint is useful to reduce the false positive decision rate especially under model uncertainty and the target is rare.

For application j , the expected resource consumption at stage i is the resource cost of feature extraction, denoted by D_i^j , weighted by the probability of the feature being selected by the previous stage. Hence,

$$\begin{aligned}E^1 &\triangleq D_1^1 + \sum_{i=1}^{K-1} D_{i+1}^1 P(\delta_i^1 = F^1) \\ E^2 &\triangleq \sum_{i=0}^{K-1} D_{i+1}^2 P(\delta_i^2 = F^2)\end{aligned}\quad (12)$$

where D_1^1 is weighted by 1 because the first-stage primary feature is always extracted. Note that D_i^j can be measured in practice by profiling the feature-extraction process.

The solution to Problem (11) is given by the following theorem.

Theorem 1. (The optimal decision rules for all applications in the cascade) For the primary application,

$$\delta_i^{1*}(\pi_i^1) = \begin{cases} 0, \pi_i^1(y_{1:i}^1) < \tau_i^{1*} \\ F^1, \text{ else} \end{cases} \quad i = 1, \dots, K-1 \quad (13)$$

$$\delta_K^{1*}(\pi_K^1) = \begin{cases} 0, \pi_K^1(y_{1:K}^1) < \tau_K^{1*} \\ 1, \text{ else} \end{cases}$$

For the secondary application,

$$\delta_0^{2*}(\pi_0^2; \pi_0^1) = F^1, \forall \pi_0^2, \forall \pi_0^1$$

$$\delta_i^{2*}(\pi_i^2; \pi_i^1) = \begin{cases} 0, \pi_i^2(y_{1:i}^2) < \tau_i^{2*}, \pi_i^1 < \tau_i^{1*} \\ F^2, \pi_i^2(y_{1:i}^2) \geq \tau_i^{2*}, \pi_i^1 < \tau_i^{1*} \\ 0, \pi_i^2(y_{1:i}^2) < \eta_i^{2*}, \pi_i^1 \geq \tau_i^{1*} \\ F^1, \pi_i^2(y_{1:i}^2) \geq \eta_i^{2*}, \pi_i^1 \geq \tau_i^{1*} \end{cases} \quad i = 1, \dots, K-1 \quad (14)$$

$$\delta_K^{2*}(\pi_K^2) = \begin{cases} 0, \pi_K^2(y_{1:K}^2) < \tau_K^{2*} \\ 1, \text{ else} \end{cases}$$

where $\tau_i^{j*}, \eta_i^{j*} \in [\pi_{L_i}^j, \pi_{U_i}^j]$ are the optimal thresholds at stage i of application j , provided that

$$C_M^2 \{ \mathbb{E}[\pi_i^2(Y_i^1)] - \mathbb{E}[\pi_i^2(Y_i^2)] \} \leq \lambda D_i^2, i = 1, \dots, K \quad (15)$$

with C_M^2 defined in Corollary 1.

Proof. See Appendix A \square

Eq. (13) in Theorem 1 shows that, for the primary application, the posterior probabilities of intermediate stages can be used to guide the execution of subsequent stages by thresholding them to decide whether to stop or extract more primary features in the next stage. The final stage has a standard detection rule, with the posterior probability being thresholded to make a prediction about the target state.

Furthermore, Eq. (14) shows that the decision rules at intermediate stages of the secondary application are not only a function of π_i^2 , but are also parameterized¹ by π_i^1 . If $\pi_i^1 \geq \tau_i^{1*}$, then according to (13), the next-stage primary feature is available, and the optimal decision always selects this feature over the secondary feature (feature sharing) for the next stage (as long as π_i^2 is above the threshold for early negative decision η_i^{2*}). Since the first-stage primary feature is always available, it is always selected by δ_0^{2*} . Otherwise, if the primary feature will not be available because $\pi_i^1 < \tau_i^{1*}$, then the secondary application falls back to selecting the secondary feature, assuming π_i^2 is above the threshold for early negative decision τ_i^{2*} . Note that the thresholds for early negative decisions are different under each case. Finally, the final stage's decision is simply a standard detection rule.

The structure of (14) favors feature-sharing whenever possible. This policy is optimal provided that additional constraints in (15) on the parameters of the cascade hold. Intuitively, (15) requires that the difference between primary and secondary feature distributions is relatively small compared to the resource cost of extracting the latter.

¹After the semicolon

Finally, the optimal threshold values $\{\tau_i^{j*}, j = 1, 2\}$, which are critical in this trade-off between performance and resource cost, can be found using Algorithm 1.

Given the above optimal decisions, Corollary 1 quantifies the optimal performance of the multi-application cascade system.

Corollary 1. (Optimal performance of applications in the cascade) For the primary application,

$$R^{1*}(\pi_0^1) \triangleq R^1(\delta_{1:K}^{1*}, \pi_0^1) = V_0^1(\pi_0^1) \quad (16)$$

where $V_0^1(\pi_0^1)$ is the result of the following recursion

$$V_K^1(\pi_K^1) \triangleq \min(\underbrace{C_M^1 \pi_K^1}_{\text{miss risk}}, \underbrace{C_A^1 (1 - \pi_K^1)}_{\text{false-alarm risk}}), \pi_K^1 \in [0, 1]$$

$$V_i^1(\pi_i^1) \triangleq \min(C_M^1 \pi_i^1, \underbrace{\lambda D_{i+1}^1 + \mathbb{E}[V_{i+1}^1(\pi_{i+1}^1(Y_{i+1}^1, \pi_i^1))]}_{\text{expected next-stage primary value function}}),$$

$$\pi_i^1 \in [\pi_{L_i}^1, \pi_{U_i}^1], i = 1, \dots, K-1$$

$$V_0^1(\pi_0^1) \triangleq \lambda D_1^1 + \mathbb{E}[V_1^1(\pi_1^1(Y_1^1, \pi_0^1))] \quad (17)$$

And the corresponding optimal thresholds are given by

$$\tau_K^{1*} = C_A^1 / (C_A^1 + C_M^1)$$

$$\tau_i^{1*} = \max(\pi_{L_i}^1, \min(\pi_{U_i}^1, \min\{\pi_i^1 : V_i^1(\pi_i^1) - C_M^1 \pi_i^1 < 0\})),$$

$$i = 1, \dots, K-1 \quad (18)$$

For the secondary application,

$$R^{2*}(\pi_0^2; \pi_0^1) \triangleq R^2(\delta_{1:K}^{2*}, \pi_0^2; \pi_0^1) = V_0^2(\pi_0^2; \pi_0^1) \quad (19)$$

where $V_0^2(\pi_0^2; \pi_0^1)$ is the result of the following recursion

$$V_K^2(\pi_K^2; \pi_i^1) \triangleq \min(C_M^2 \pi_K^2, C_A^2 (1 - \pi_K^2)), \pi_K^2 \in [0, 1]$$

$$V_i^2(\pi_i^2; \pi_i^1) \triangleq \begin{cases} \min(C_M^2 \pi_i^2, \underbrace{\lambda D_{i+1}^2 + \mathbb{E}[V_{i+1}^2(\pi_{i+1}^2(Y_{i+1}^2, \pi_i^2); \pi_i^1)]}_{\text{expected next-stage secondary value function using the secondary feature}}), \\ \text{if } \pi_i^1 < \tau_i^{1*} \\ \min(C_M^2 \pi_i^2, \underbrace{\mathbb{E}[V_{i+1}^2(\pi_{i+1}^2(Y_{i+1}^1, \pi_i^2); \pi_i^1)]}_{\text{expected next-stage secondary value function using the shared primary feature}}), \\ \text{else} \end{cases}$$

$$\pi_i^2 \in [\pi_{L_i}^2, \pi_{U_i}^2], i = 1, \dots, K-1$$

$$V_0^2(\pi_0^2; \pi_0^1) \triangleq \mathbb{E}[V_1^2(\pi_1^2(Y_1^1, \pi_0^2); \pi_0^1)] \quad (20)$$

and the corresponding optimal thresholds are given by

$$\tau_K^{2*} = C_A^2 / (C_A^2 + C_M^2)$$

$$\tau_i^{2*} = \max(\pi_{L_i}^2, \min(\pi_{U_i}^2, \min\{\pi_i^2 : V_i^2(\pi_i^2; \pi_i^1) - C_M^2 \pi_i^2 < 0\})),$$

$$\text{if } \pi_i^1 < \tau_i^{1*} \quad (21)$$

$$\eta_i^{2*} = \max(\pi_{L_i}^2, \min(\pi_{U_i}^2, \min\{\pi_i^2 : V_i^2(\pi_i^2; \pi_i^1) - C_M^2 \pi_i^2 < 0\})),$$

$$\text{else ,}$$

$$i = 1, \dots, K-1,$$

where $C_M^j, C_A^j, j = 1, 2$ are the costs associated with miss and false-alarm decisions of application j .

Corollary 1 shows that the optimal performance achieved by each application can be found using a recursive procedure. The procedure has K iterations, each corresponding to a stage in the system. Starting from the last stage K and proceeding backward to 0, the value function V_i^j is recursively updated (see (17) and (20)). The last-stage value function V_K^j is the minimum of the miss and false-alarm risks across π_K . An intermediate-stage value function $V_i^j, i = 1, \dots, K - 1$ is the minimum of the miss risk and the *expected next-stage* value function, which requires the probabilistic updates in (5),(6). The final value function V_0^j is the minimal risk achievable by an application.

Note that the secondary application's value function at an intermediate stage is given by different expressions depending on the availability of the primary feature for the next stage, i.e. $\pi_i^1 \leq \tau_i^{1*}$ (see (20)). If the primary feature is not available for the next stage, then the expected next-stage value function is taken with respect to the secondary feature, whose extraction cost (λD_{i+1}^2) is also included. Otherwise, the expected next-stage value function does not contain the resource cost to extract the secondary feature and the expectation is taken with respect to the primary feature.

Once a value function is known, then the corresponding optimal threshold can be found using just arithmetic operations, i.e. comparing the value function with the miss risk (see (18) and (21)). For the last stage K , the optimal threshold can be given in closed form. Note that the intermediate stages' thresholds are capped between the upper and lower bounds due to model uncertainty (see Section III-A). For the secondary application, depending on the availability of the primary feature, the thresholds for early negative decisions are different, and hence denoted differently (τ_i^2 and η_i^2 , respectively). Intuitively, if the primary threshold is low, i.e. the primary application consumes most of the resource budget, then the secondary application is more inclined to use the shared primary feature due to low resource budget. On the other hand, if the primary threshold is high, i.e. the secondary application has most of the resource budget, then the secondary feature is used more to reduce its miss and false-alarm risks.

The discussion so far has been focusing on optimizing parameters of the cascade design. A natural next question to ask is whether the constraints of the cascade design can be relaxed to further improve performance. Namely, would introducing additional degrees of freedom, i.e. early positive decisions in intermediate stages, to the cascade *always* improve its performance? Intuitively, when model uncertainties of intermediate stages are accounted for (see Section III-A), and it is known *a priori* that the target is rare, early positive decisions are likely to have higher risk and hence are discouraged. Therefore, introducing additional early positive decisions does *not* always improve the performance of the cascade. The precise conditions for which the cascade design itself is optimal is given by the following proposition.

Proposition 1. (*Optimality of the cascade design*) *With model uncertainty, introducing additional early positive decisions in*

intermediate stages of the cascade does not improve performance, as long as

$$\underbrace{\max\{\pi_i^j : V_i^j(\pi_i^j) - C_A^j(1 - \pi_i^j) < 0\}}_{\text{optimal threshold for early positive decision}} > \pi_{U_i}^j, \quad (22)$$

$$i = 1, \dots, K - 1, j = 1, 2$$

Proof. See Appendix B. \square

The left-hand side of (22) is the optimal threshold corresponding to an early positive decision. Namely, these additional decisions also have threshold-based optimal policies (see Appendix B), and a posterior probability *above* such a threshold shall trigger an early positive decision. If such a threshold is above the upper bound on the posterior probability at a stage, then its early positive decision is never selected, and hence does not affect the performance of the cascade.

IV. SYSTEM SIMULATION

This section applies the theory developed in Section III to design a multi-application, acoustic detection system.

A. Hardware components

The hardware components needed for an acoustic sensing system are listed in Table I, along with their power consumption (from the datasheets). Note that these are commercially-off-the-shelf (COTS) components, without any customization. The sensor's brain (supplied at 3.6 V) is Silicon Labs's WGM110, which is a low-power wireless (wifi) chip that includes a low-power 12-bit ADC, an ARM Cortex-M3 processor, and a wifi module (among others). All the control logic and the (digital) signal processing software are assumed to be implemented on this general-purposed processor, without any ASIC² or DSP³. In addition, a microphone and a preamp are also a part of the acoustic sensor. The power consumption of the microphone, the preamp circuit, and the ADC, altogether is 3.6 mW and considered as the baseline of the system. Data collected by the sensor are transmitted downstream to the client, which is a ML100G-10 Next Unit of Computing (NUC) from LogicSupply. Power profiling the NUC (using PowerBlade [26]) results in an average power consumption of 4.744 W (at 9 V).

TABLE I: Power consumption of hardware components of the acoustic sensing system.

Components	Power consumption (mW)
Electret Microphone	0.72
Preamp Circuit (OPA344)	1.08
WGM110 12-bit ADC	1.8 ^a
WGM110 ARM core	86.4
WGM110 transmission	900
ML100G-10	4744

²application-specific integrated circuit

³digital signal processor

^aFrom [22], the ADC draws 0.5 mA, which is equivalent to 1.8 mW at 3.6 V.

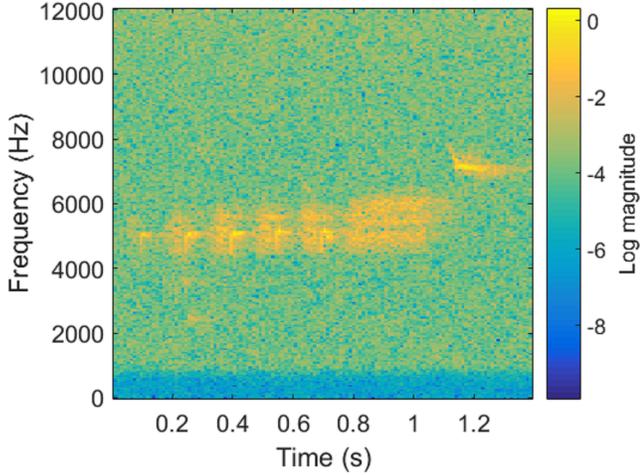


Fig. 2: Spectrogram of a sample GCW's (type-A) call.

B. Software components

1) *Primary application (Golden-Cheeked Warbler detection)*: The detection of the Golden-cheeked Warbler (GCW)'s (type-A) calls [27] is considered as the primary application. Namely, $X^1 = 1$ indicates the presence of a GCW call, and $X^1 = 0$ otherwise. Since the GCW is an endangered bird species, this application has important implications for their conservation.

The application's software is organized into three subtasks: generic energy-based analysis, spectral-based analysis, and temporal-spectral-based analysis. The energy analysis is a low-complexity computation that produces energy-based features useful for detecting acoustic events from silence. The spectral-based analysis takes into account the spectral information about the GCW calls, which only has energy in the 4500-6500 Hz and 7000-8000 Hz bands (see Fig. 2), to produce band-specific, energy-based features using standard DSP filtering techniques. Finally, the spectral-temporal-based analysis takes into account both the spectral and temporal structure of the GCW call from Fig. 2 to produce reliable, indicative features using a template matching technique. Note that the input into the above analyses is an audio stream (or precisely, its high-dimensional time-frequency representation, see Fig. 2), and their output is a scalar score sequence, i.e. a score for each audio frame. Hence, these analyses effectively perform dimensionality reduction.

Since the generic energy analysis has low computational complexity and can help prune out a significant amount of noise-only data from the audio stream early, it is executed on edge/sensor nodes. Only acoustic events are transmitted downstream to clients, where spectral and temporal-spectral-based analyses are further carried out. The system diagram is illustrated in Figure 3 and arranged to fit the proposed cascade abstraction. Note that the physical separation (between sensors and clients) does not necessarily correspond to the logical separation (between stages). For instance, the cost of data transmission on sensors are included into the cost of executing the second stage, along with the cost of spectral-

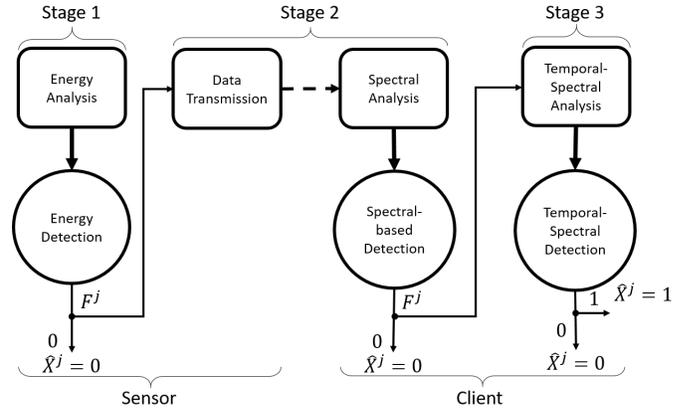


Fig. 3: The software components of the primary application is organized as a cascade with 3 stages: energy analysis as stage 1, spectral-based analysis (along with the data transmission) as stage 2, and temporal-spectral analysis as stage 3. Note that components of the cascade are implemented distributedly across the network, with the dashed line representing a remote connection.

based analysis on clients, since they are both a result of the first-stage decision.

The resource cost parameters at each stage D_i^1 , $i = 1, 2, 3$, which can be estimated from values of Table I and the execution times of the software components, are needed to optimize the resource-performance trade-off. It is assumed that all processing finishes before a periodic deadline, i.e. when buffers (an ADC buffer on the sensor, a task buffer on the client) are full. The average execution time of each task (per audio frame of 32 ms) can be estimated/profiled and is given as follows. The energy analysis takes 16 ms⁴. The average transmission time takes 11 ms (500 ms for a 1.5 s event⁵). Finally, the spectral and temporal-spectral analyses take 0.37 μ s and 15 ms, respectively⁶. Hence,

$$\begin{aligned} D_1^1 &= 86.4 \times 0.016, \\ D_2^1 &= 900 \times 0.011 + 4744 \times 0.37 \times 10^{-6}, \\ D_3^1 &= 4744 \times 0.015, \end{aligned} \quad (23)$$

Our dataset is a 46-minute, 24 kHz audio recording at the field in Rancho Diana, San Antonio's city park. The dataset contains 206 GCW calls (manually identified and labeled), each of whose duration is approximately one second. In addition to GCW calls, the dataset also contains various interferences from other animals' vocalization, time-varying wind noise, etc., since it is taken directly from field recording. Precisely, the fraction of GCW calls in the entire dataset is 10.19%. Hence, this detection problem belongs to the rare-target class, where the prior is asymmetrical, i.e. $\pi_0^1 \ll 0.5$. In this simulation, we consider a range of prior in the rare-event regime, i.e. $\pi_0^1 \in [0.05, 0.20]$. Finally, the miss and false-alarm

⁴Estimated as half of the frame length.

⁵Profiled on a prototype.

⁶Profiled in MatLab for ML100G-10.

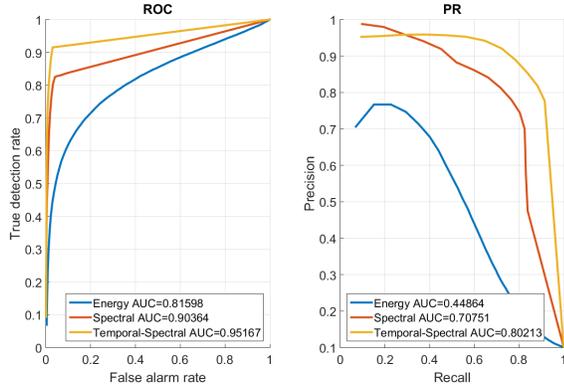


Fig. 4: Receiver operating characteristic (ROC) curves and precision-recall (PR) curves of the features produced by the 3 analyses.

costs are given by $C_M^1 = 2$, $C_A^1 = 1$ to emphasize that the miss risk is higher in this setting.

The dataset are input to each of the three analyses discussed above. The scalar output scores from each analysis are taken as its respective features, resulting in a total of three feature sets/groups/types. The discriminative power of each feature type, or equivalently the performance of an analysis, can be quantified using receiver operating characteristic (ROC) and precision-recall (PR) curves as shown in Fig. 4. From the figure, it is evident that the temporal-spectral feature is better than the spectral feature, which in turn is better than the generic energy feature, at detecting GCW calls.

The conditional probability mass functions (PMF), i.e. $p_i(y_i|x)$, of features from each analysis can be estimated up to some quantization level, i.e. 100. Furthermore, as alluded to in Section III-A, energy-based and spectral-based features, by construction, are inadequate to characterize GCW calls, and hence there are inherent uncertainties in these features for the detection of GCW calls. These uncertainties can be explicitly accounted for in the features' distributions using the uncertainty model discussed in Section III-A, with the following parameters.

$$\begin{aligned}
 \epsilon_{01}^1 &= \epsilon_{02}^1 = 0.1 \\
 \epsilon_{11}^1 &= \epsilon_{12}^1 = 0.1 \\
 \nu_{01}^1 &= \nu_{02}^1 = 0.1 \\
 \nu_{11}^1 &= \nu_{12}^1 = 0.1
 \end{aligned} \tag{24}$$

Intuitively, the ϵ and the ν parameters indicate the level and the strength of a contamination on the nominal distribution, respectively. A formal method to set these parameters are left for future work. Finally, it is assumed that the temporal-spectral analysis (the last stage) is sufficient to characterize GCW calls and hence there is no uncertainty in this feature set.

2) *Secondary application:* To illustrate the benefit of feature sharing, we invoke the following twin-comparison argument. We consider a hypothetical secondary application that is, as far as the resource-performance trade-off is concerned, identical to the primary application. Namely, all parameters,

such as the resource cost and the feature models at each stage, of the secondary application is the same as those of the primary one. Due to the asymmetry in feature sharing between the primary and secondary applications, it is expected that there will be differences in the resulting resource consumption and detection performance of the two applications, and the merit of the proposed feature sharing approach can be evaluated by quantifying this difference.

C. Results

The method developed in Section III can be used to optimize the acoustic system and the results are presented below.

Fig. 5 breaks down the primary application's risk into the weighted resource consumption, and the miss and false-alarm rates to provide an intuitive understanding of the optimal policies. Furthermore, the average resource consumption of the primary application across all priors is 44.406 mJ (per audio frame). Without feature sharing, it is expected that the resource consumption of the secondary application would be the same as that of the primary one. However, the average resource consumption of the secondary application across the priors of interest can be found to be 4.877 mJ (based on the break-down of the secondary risk illustrated in Fig. 6), which is approximately a $9\times$ reduction in resource consumption. This significant resource-saving is due to the fact that extracted features are shared and not recomputed among applications. In addition, the average detection risk (both the miss and false-alarm rate) of the secondary application is also reduced by $1.43\times$ (from 4.75% to 3.31%). This reduction in risk illustrates that using shared features can be more beneficial than having no feature at all.

It is worth noting that the above applications' resource consumptions are the result of setting the regularization parameter λ to 0.0043, which is optimal for a resource/energy budget of 49.398 mJ (the sum of power consumptions by both applications and the 3.6×0.032 mJ base line from the microphone, the preamp circuit, and the ADC of the sensor over a frame). For a given resource budget, one needs to be solved for λ . Numerical solution of the scalar variable λ is straightforward and hence shall be skipped here.

The primary and secondary decision rules are illustrated in Fig. 7 and 8, respectively. Note that the optimal policies of the secondary application take advantage of feature sharing whenever possible. Namely, $\delta_i^{2*} = F^1$ for all π_i^2 and $\pi_i^1 \geq \tau_i^{1*}$, $i = 0, 1, 2$. When the primary feature is not available, the secondary policies choose between extracting the secondary feature ($\delta_i^{2*} = F^2$) or making an early negative decision ($\delta_i^{2*} = 0$).

V. CONCLUSION

This paper investigates and shows the benefits of features sharing in the optimization of resource-performance trade-off for detection systems with multiple applications. The proposed system model focuses on the vertical design, rather than the horizontal one commonly seen in the wireless sensor network literature. Namely, it is assumed that there is only one device at each layer/stage in the system stack, as opposed

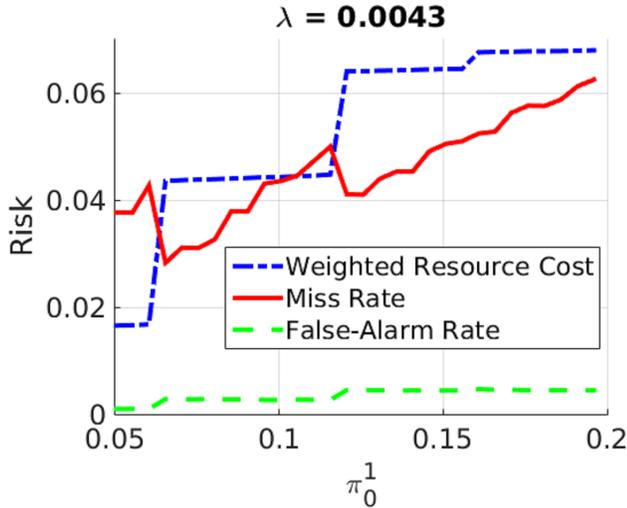


Fig. 5: Breakdown of the optimal primary risk into components (see Eq. (11)): false negative (miss), false positive (false-alarm), and Lagrangian-weighted resource consumption. Low false-alarm rate is achieved across the priors of interest. The miss rate tends to increase with the prior. At a certain level, the primary application must ramp up its resource consumption or incur more false-alarm to reduce the miss rate.

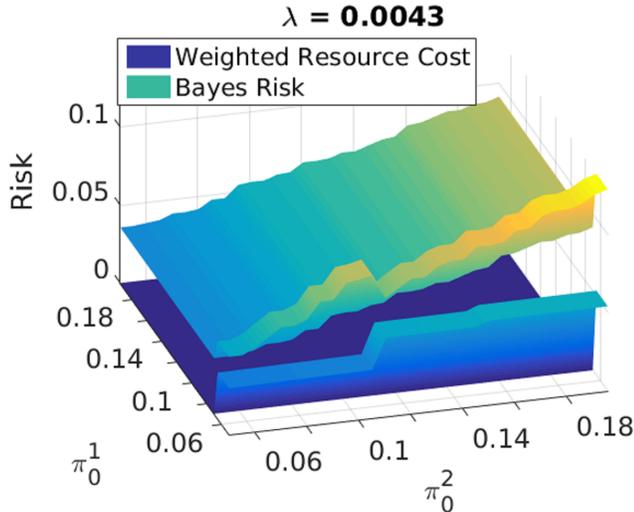


Fig. 6: Breakdown of the optimal secondary risk into components (see Eq. (11)): Detection risk and Lagrangian-weighted resource consumption. The detection risk tends to increase with the secondary prior. At a certain level, the secondary application must ramp up its resource consumption to reduce the risk.

to having multiple devices at the same layer. Therefore, this work can complement prior works and vice versa. For instance, Chamberland et al. showed that it is asymptotically optimal for all sensors in a power-constrained sensor network to adopt the same (transmission) strategy, as the number of sensors in the network goes to infinity [28]. The result in [28] therefore allows ours to be applicable for layers with more than one device. Finally, a proof-of-concept implementation

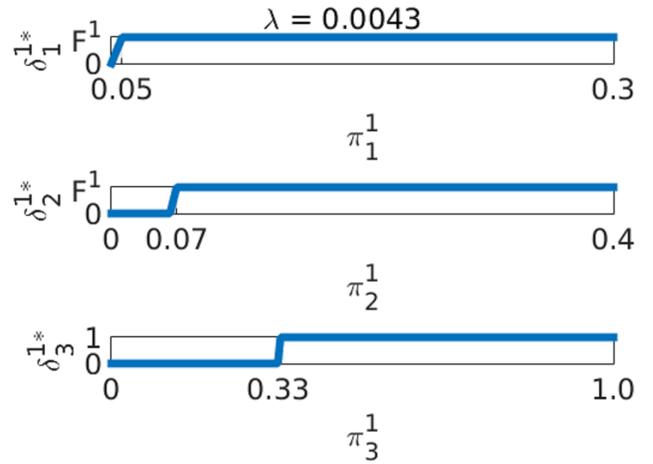


Fig. 7: Optimal decision rules of the primary application $\delta_i^{1*}(\pi_i^1) \in \{F^1, 0, 1\}, i = 1, \dots, 3$.

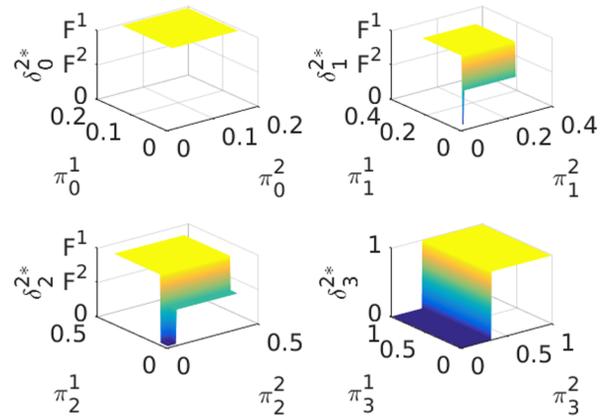


Fig. 8: Optimal decision rules of the secondary application $\delta_i^{2*}(\pi_i^{1:2}) \in \{F^1, F^2, 0, 1\}, i = 0, \dots, 3$.

of the acoustic system in Section IV (with the sensor as an Android app) is available online for demonstration⁷.

ACKNOWLEDGEMENTS

This work was supported in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, and in part by a research grant for the Human-Centered Cyberphysical Systems Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR)

APPENDIX

A. Proof of Theorem 1

We start by expanding the risk terms in (11). The false negative (miss) rate due to early negative decision for the first

⁷At <http://acoustic.ifp.illinois.edu>

stage is

$$\begin{aligned} R_{1,M}^1 &= \int p(dy_1^1) \{C_M^1 \pi_1^1(y_1^1) \mathbb{I}(\delta_1^1 = 0)\} \\ R_{1,M}^2 &= \int p(dy_{1:2}^1) \{C_M^2 \pi_1^2(y_1^2) \mathbb{I}(\delta_1^2 = 0, \delta_0^2 = F^2) + \\ &\quad C_M^2 \pi_1^2(y_1^1) \mathbb{I}(\delta_1^2 = 0, \delta_0^2 = F^1)\} \end{aligned} \quad (25)$$

where $R_{1,M}^1, R_{1,M}^2$ are the first-stage miss risk of the primary and secondary applications, respectively. Furthermore, the first term of $R_{1,M}^2$ is due to using the secondary feature y_1^2 ($\delta_0^2 = F^2$), and the second term is due to using the shared (primary) feature y_1^1 ($\delta_0^2 = F^1$). $\mathbb{I}(\cdot)$ denotes the indicator function that takes value 1 when its argument (a probability event) is true and 0 otherwise. Finally, $p(dy_{1:K})$ is the probability measure of feature realizations $y_{1:K}$.

Likewise, the miss terms for the stage $i = 2, \dots, K$ can be given as follows.

$$\begin{aligned} R_{i,M}^1 &= \int p(dy_{1:i}^1) \{C_M^1 \pi_i^1(y_{1:i}^1) \mathbb{I}(\delta_i^1 = 0, \delta_{i-1}^1 = F^1)\} \\ R_{i,M}^2 &= \int p(dy_{1:i}^{1:2}) \{C_M^2 \pi_i^2(y_{1:i-1}^1, y_i^2) \mathbb{I}(\delta_i^2 = 0, \delta_{i-1}^2 = F^2) + \\ &\quad C_M^2 \pi_i^2(y_{1:i-1}^1, y_i^1) \mathbb{I}(\delta_i^2 = 0, \delta_{i-1}^2 = F^1, \delta_{i-1}^1 = F^1)\} \end{aligned} \quad (26)$$

where the first term of $R_{i,M}^2$ is again due to using the secondary feature y_i^2 ($\delta_{i-1}^2 = F^2$), and the second term is due to using the shared feature y_i^1 ($\delta_{i-1}^2 = F^1$ and $\delta_{i-1}^1 = F^1$). Similarly, the false-alarm (false positive) term at the last stage is given as follows.

$$\begin{aligned} R_{K,A}^1 &= \int p(dy_{1:K}^1) \{C_A^1 (1 - \pi_K^1(y_{1:K}^1)) \\ &\quad \mathbb{I}(\delta_K^1 = 1, \delta_{K-1}^1 = F^1)\} \\ R_{K,A}^2 &= \int p(dy_{1:K}^{1:2}) \{C_A^2 (1 - \pi_K^2(y_{1:K-1}^1, y_K^2)) \\ &\quad \mathbb{I}(\delta_K^2 = 1, \delta_{K-1}^2 = F^2) + \\ &\quad C_A^2 (1 - \pi_K^2(y_{1:K-1}^1, y_K^1)) \\ &\quad \mathbb{I}(\delta_K^2 = 1, \delta_{K-1}^2 = F^1, \delta_{K-1}^1 = F^1)\} \end{aligned} \quad (27)$$

An important step in solving Problem (11) is the following expansion of the expected resource cost in (12). By the law of total probability,

$$\begin{aligned} D_1^1 &= D_1^1 \left\{ P(\delta_1^1 = 0) + \sum_{i=2}^{K-1} P(\delta_i^1 = 0, \delta_{i-1}^1 = F^1) + \right. \\ &\quad \left. P(\delta_K^1 = 0, \delta_{K-1}^1 = F^1) + P(\delta_K^1 = 1, \delta_{K-1}^1 = F^1) \right\} \end{aligned} \quad (28)$$

and

$$\begin{aligned} D_{i+1}^1 P(\delta_i^1 = F^1) &= D_{i+1}^1 \left\{ \sum_{j=i+1}^{K-1} P(\delta_j^1 = 0, \delta_{j-1}^1 = F^1) + \right. \\ &\quad \left. P(\delta_K^1 = 0, \delta_{K-1}^1 = F^1) + P(\delta_K^1 = 1, \delta_{K-1}^1 = F^1) \right\}, \\ &\quad i = 1, \dots, K-1 \end{aligned} \quad (29)$$

Similarly for the secondary application

$$\begin{aligned} D_{i+1}^2 P(\delta_i^2 = F^2) &= D_{i+1}^2 \left\{ \sum_{j=i+1}^{K-1} P(\delta_j^2 = 0, \delta_{j-1}^2 = F^2) + \right. \\ &\quad \left. P(\delta_K^2 = 0, \delta_{K-1}^2 = F^2) + P(\delta_K^2 = 1, \delta_{K-1}^2 = F^2) \right\}, \\ &\quad i = 0, \dots, K-1 \end{aligned} \quad (30)$$

Putting everything back into (11) yields a dynamic programming structure, with the state variable being the posteriors $\pi_i^j, j = 1, 2$ defined in Section III-A. Minimizing (11) can thus be achieved efficiently using the following backward procedure.

$$\begin{aligned} V_K^1(\pi_K^1) &\triangleq \min_{\delta_K^1} \mathbb{I}(\delta_K^1 = 0) C_M^1 \pi_K^1 + \mathbb{I}(\delta_K^1 = 1) C_A^1 (1 - \pi_K^1) \\ V_K^2(\pi_K^2) &\triangleq \min_{\delta_K^2} \mathbb{I}(\delta_K^2 = 0) C_M^2 \pi_K^2 + \mathbb{I}(\delta_K^2 = 1) C_A^2 (1 - \pi_K^2) \\ V_i^1(\pi_i^1) &\triangleq \min_{\delta_i^1} \mathbb{I}(\delta_i^1 = 0) C_M^1 \pi_i^1 + \\ &\quad \mathbb{I}(\delta_i^1 = F^1) \left\{ \lambda D_{i+1}^1 + \mathbb{E}[V_{i+1}^1(\pi_{i+1}^1(Y_{i+1}^1, \pi_i^1))] \right\} \\ V_i^2(\pi_i^2; \pi_i^1) &\triangleq \min_{\delta_i^2} \mathbb{I}(\delta_i^2 = 0) C_M^2 \pi_i^2 + \\ &\quad \mathbb{I}(\delta_i^{1*} = F^1, \delta_i^2 = F^1) \mathbb{E}[V_{i+1}^2(\pi_{i+1}^2(Y_{i+1}^1, \pi_i^2); \pi_i^1)] + \\ &\quad \mathbb{I}(\delta_i^2 = F^2) \left\{ \lambda D_{i+1}^2 + \mathbb{E}[V_{i+1}^2(\pi_{i+1}^2(Y_{i+1}^2, \pi_i^2); \pi_i^1)] \right\} \\ &\quad i = 1, \dots, K-1 \\ V_0^1(\pi_0^1) &\triangleq \lambda D_1^1 + \mathbb{E}[V_1^1(\pi_1^1(Y_1^1, \pi_0^1))] \\ V_0^2(\pi_0^2; \pi_0^1) &\triangleq \min_{\delta_0^2} \mathbb{I}(\delta_0^2 = F^1) \mathbb{E}[V_1^2(\pi_1^2(Y_1^1, \pi_0^2); \pi_0^1)] + \\ &\quad \mathbb{I}(\delta_0^2 = F^2) \left\{ \lambda D_1^2 + \mathbb{E}[V_1^2(\pi_1^2(Y_1^2, \pi_0^2); \pi_0^1)] \right\} \end{aligned} \quad (31)$$

where the expectation is taken with respect to the evidence probabilities (see Section III-A) and V_i^j is the value function at stage i of application j . From the first and third expressions of (31), the minimizers for the primary application can be obtained by setting

$$\delta_K^{1*}(\pi_K^1) = \begin{cases} 0, & \pi_K^1 < C_A^1 / (C_A^1 + C_M^1) \\ 1, & \text{else} \end{cases} \quad (32)$$

and

$$\delta_i^{1*}(\pi_i^1) = \begin{cases} 0, & V_i^1(\pi_i^1) = C_M^1 \pi_i^1 \\ F^1, & V_i^1(\pi_i^1) < C_M^1 \pi_i^1 \end{cases}, \quad (33)$$

$$i = 1, \dots, K-1$$

The expression in (33) can be further simplified into (13) using Lemmas 1.1 and 1.3.

From the second and fourth expressions of (31), the optimal decision rule for the secondary application is

$$\delta_K^{2*}(\pi_K^2) = \begin{cases} 0, & \pi_K^2 < C_A^2 / (C_A^2 + C_M^2) \\ 1, & \text{else} \end{cases} \quad (34)$$

and

$$\delta_i^{2*}(\pi_i^2; \pi_i^1) = \begin{cases} 0, V_i^2 = C_M^2 \pi_i^2, \\ F^2, V_i^2 = \lambda D_{i+1}^2 + \mathbb{E}[V_{i+1}^2(Y_{i+1}^2, \pi_i^2; \pi_i^1)] \\ F^1, V_i^2 = \mathbb{E}[V_{i+1}^2(Y_{i+1}^1, \pi_i^2; \pi_i^1)], \pi_i^1 \geq \tau_i^{1*} \end{cases},$$

$$i = 0, \dots, K-1 \quad (35)$$

The expression in (35) can be further simplified into (14) using Lemma 1.4.

Lemma 1.1. $\mathbb{E}[V_{i+1}(\pi_{i+1}(Y_{i+1}, \pi))], i = 0, \dots, K-1$ and $V_i(\pi), i = 1, \dots, K$ are concave⁸.

Proof. $V_K(\pi)$ is concave. Hence, by Lemma 1.2, $\mathbb{E}[V_K(\pi_K(Y_K, \pi))]$ is concave.

Assume that $V_{i+1}(\pi)$ is concave, thus $\mathbb{E}[V_{i+1}(\pi_{i+1}(Y_{i+1}, \pi))]$ is concave by Lemma 1.2, then

$$V_i(\pi) = \min\{(\pi), \lambda D_{i+1} + \mathbb{E}[V_{i+1}(\pi_{i+1}(Y_{i+1}, \pi))]\} \quad (36)$$

is also concave. Again, by Lemma 1.2, $\mathbb{E}[V_i(\pi_i(Y_i, \pi))]$ is concave. \square

Lemma 1.2. $\mathbb{E}[V_{i+1}(\pi_{i+1}(Y_{i+1}, \pi))]$ is concave if $V_{i+1}(\pi)$ is concave.

Proof. See [30, p. 146]. \square

Lemma 1.3. $\mathbb{E}[V_{i+1}(\pi_{i+1}(Y_{i+1}, 0))] = 0, i = 0, \dots, K-1$.

Proof. $V_K(0) = 0$, then $\mathbb{E}[V_K(\pi_K(Y_K, 0))] = V_K(0) = 0$

Assume that $\mathbb{E}[V_{i+1}(\pi_{i+1}(Y_{i+1}, 0))] = 0$, then

$$V_i(0) = \min\{0, \lambda D_{i+1}\} = 0. \quad (37)$$

Hence, $\mathbb{E}[V_i(\pi_i(Y_i, 0))] = V_i(0) = 0$. \square

Lemma 1.4. If the condition in (15) holds, then

$$\delta_i^{2*}(\pi_i^2; \pi_i^1) = \begin{cases} 0, V_i^2 = \pi_i^2, \pi_i^1 < \tau_i^{1*} \\ F^2, V_i^2 < \pi_i^2, \pi_i^1 < \tau_i^{1*} \\ 0, V_i^2 = \pi_i^2, \pi_i^1 \geq \tau_i^{1*} \\ F^1, V_i^2 < \pi_i^2, \pi_i^1 \geq \tau_i^{1*} \end{cases}, \quad (38)$$

$$i = 0, \dots, K-1$$

which implies $\delta_i^{2*} \neq F^2$ when $\pi_i^1 \geq \tau_i^{1*}$.

Proof. The fourth expression of (31) is equivalent to Eq. (38) if and only if

$$\mathbb{E}[V_i^2(Y_i^1, \pi_{i-1}^2)] - \mathbb{E}[V_i^2(Y_i^2, \pi_{i-1}^2)] \leq \lambda D_i^2 \quad (39)$$

The condition in (39) is made satisfied by (15) because of Lemma 1.5 (note that V_i^2 is concave over π_i^2 for each $\pi_i^1, i = 1, \dots, K$). \square

Lemma 1.5.

$$\mathbb{E}[V_i(Y_i^1, \pi_{i-1}^1)] - \mathbb{E}[V_i(Y_i^2, \pi_{i-1}^1)] \leq C_M^2 \{ \mathbb{E}[\pi_i(Y_i^1)] - \mathbb{E}[\pi_i(Y_i^2)] \}$$

$$i = 1, \dots, K \quad (40)$$

⁸Moreover, $V_i(\pi), i = 1, \dots, K$ can be shown to be piece-wise linear and concave, which was first observed and proven (by induction) in [29, Smallwood and Sondik].

Proof. Since $V_i(\pi_i)$ is concave, $V_i'(\pi_i)$ is non-increasing. Furthermore, $V_i'(\epsilon) = C_M^2$ for some small $\epsilon > 0$. Therefore, $V_i'(\pi_i) \leq C_M^2$, i.e.

$$V_i(\pi_i(Y_i^1)) - V_i(\pi_i(Y_i^2)) \leq C_M^2 [\pi_i(Y_i^1) - \pi_i(Y_i^2)] \quad (41)$$

Taking expectation on both size of (41) yields (40). \square

B. Proof of Proposition 1

Introducing (additional) early positive decisions to intermediate stages results in the following modification to the third and fourth lines of (31).

$$V_i^1(\pi_i^1) \triangleq \min_{\delta_i^1} \mathbb{I}(\delta_i^1 = 0) C_M^1 \pi_i^1 + \mathbb{I}(\delta_i^1 = 1) C_A^1 (1 - \pi_i^1)$$

$$\mathbb{I}(\delta_i^1 = F^1) \left\{ \lambda D_{i+1}^1 + \mathbb{E}[V_{i+1}^1(\pi_{i+1}^1(Y_{i+1}^1, \pi_i^1))] \right\}$$

$$V_i^2(\pi_i^2; \pi_i^1) \triangleq \min_{\delta_i^2} \mathbb{I}(\delta_i^2 = 0) C_M^2 \pi_i^2 + \mathbb{I}(\delta_i^2 = 1) C_A^2 (1 - \pi_i^2)$$

$$\mathbb{I}(\delta_i^{1*} = F^1, \delta_i^2 = F^1) \mathbb{E}[V_{i+1}^2(\pi_{i+1}^1(Y_{i+1}^1, \pi_i^2); \pi_i^1)] +$$

$$\mathbb{I}(\delta_i^2 = F^2) \left\{ \lambda D_{i+1}^2 + \mathbb{E}[V_{i+1}^2(\pi_{i+1}^2(Y_{i+1}^2, \pi_i^2); \pi_i^1)] \right\}$$

$$i = 1, \dots, K-1 \quad (42)$$

Therefore the positive decision is *not* chosen by the optimal policy under the following circumstances.

$$\delta_i^{j*} \neq 1 \text{ if } V_i^j < C_A^j (1 - \pi_i^j),$$

$$i = 1, \dots, K-1, j = 1, 2 \quad (43)$$

Since V_i^1 and V_i^2 are concave functions of π_i^1 and π_i^2 , respectively, (43) is equivalent to

$$\delta_i^{j*} \neq 1 \text{ if } \pi_i^j \leq \max\{\pi_i^j : V_i^j < C_A^j (1 - \pi_i^j)\},$$

$$i = 1, \dots, K-1, j = 1, 2 \quad (44)$$

Hence if (22) holds then the positive decisions are never chosen by the optimal policy, and therefore do not make any difference in the end system performance.

REFERENCES

- [1] E. A. Lee, J. D. Kubiawicz, J. M. Rabaey, A. L. Sangiovanni-Vincentelli, S. A. Seshia, J. Wawrzyniek, D. Blaauw, P. Dutta, K. Fu, C. Guestrin *et al.*, "The TerraSwarm Research Center (TSRC)(A white paper)," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2012-207*, 2012.
- [2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [3] L. Le, D. M. Jun, and D. L. Jones, "Energy-efficient detection system in time-varying signal and noise power," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013. IEEE, 2013, pp. 2736–2740.
- [4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1. IEEE, 2001, pp. I–511.
- [5] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 1041–1044.
- [6] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.
- [7] D. S. Turaga, O. Verscheure, U. V. Chaudhari, and L. D. Amini, "Resource management for networked classifiers in distributed stream mining systems," in *Sixth International Conference on Data Mining*, 2006. IEEE, 2006, pp. 1102–1107.

Algorithm 1 Pseudo-code to find optimal thresholds for the multi-application cascade system. This algorithm has the time complexity of $O(KM^2L)$ and the space complexity of $\max\{O(KM^2), O(M^2L)\}$, where L is the quantization level of the feature models.

```

1: function OPTIMIZE( $model^1, model^2$ )
2:    $model^1, model^2$  are structures containing the primary
   and secondary application's feature models, respectively
3:    $K$  is the number of stages
4:    $M$  is the state probability quantization size
5:   Use (7) to obtain robust versions of  $model^1$  and
    $model^2$ .
6:    $b = [0 : 1/(M-1) : 1]$  (dummy) probability vector
7:    $V_K^1 = \min(C_M^1 b, C_A^1(1-b))$ 
8:    $\tau_K^{1*} = C_A^1 / (C_A^1 + C_M^1)$ 
9:   for  $i = 1 : 1 : M$  do
10:     $V_K^2(:, i) = \min(C_M^2 b, C_A^2(1-b))$ 
11:     $\tau_K^{2*}(i) = C_A^2 / (C_A^2 + C_M^2)$ 
12:   end for
13:   for  $i = K-1 : -1 : 1$  do
14:     $J^1 =$  expected next-stage ( $i+1$ ) primary value
   function
15:     $V_i^1 = \min(C_M^1 b, J^1)$ 
16:     $\tau_i^{1*} = \min\{b : V_i^1 - C_M^1 b < 0\}$ 
17:     $\tau_i^{1*} = \max(\pi_{L_i}^1, \min(\pi_{U_i}^1, \tau_i^{1*}))$ 
18:     $J^{21} =$  expected next-stage ( $i+1$ ) secondary value
   function using the shared primary feature
19:     $J^{22} =$  expected next-stage ( $i+1$ ) secondary value
   function using the secondary feature
20:    for  $j = 1 : 1 : M$  do
21:      if  $b(j) < \tau_i^{1*}$  then
22:         $V_i^2(:, j) = \min(C_M^2 b, J^{22}(:, j))$ 
23:         $\tau_i^{2*}(j) = \min\{b : V_i^2(:, j) - C_M^2 b < 0\}$ 
24:         $\tau_i^{2*}(j) = \max(\pi_{L_i}^2, \min(\pi_{U_i}^2, \tau_i^{2*}(j)))$ 
25:      else
26:         $V_i^2(:, j) = \min(C_M^2 b, J^{21}(:, j))$ 
27:         $\eta_i^{2*}(j) = \min\{b : V_i^2(:, j) - C_M^2 b < 0\}$ 
28:         $\eta_i^{2*}(j) = \max(\pi_{L_i}^2, \min(\pi_{U_i}^2, \eta_i^{2*}(j)))$ 
29:      end if
30:    end for
31:  end for
32:   $V_0^1 = J^1 =$  expected next-stage (1) primary value
   function
33:   $V_0^2 = J^{21} =$  expected next-stage (1) secondary value
   function using the shared primary feature
34: end function

```

- [8] Z.-B. Tang, K. R. Pattipati, and D. L. Kleinman, "Optimization of detection networks: Part I - Tandem structures," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, no. 5, pp. 1044–1059, 1991.
- [9] P. F. Swaszek, "On the performance of serial networks in distributed detection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, no. 1, pp. 254–260, 1993.
- [10] R. Viswanathan, S. C. Thomopoulos, and R. Tumuluri, "Optimal serial distributed decision fusion," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 4, pp. 366–376, 1988.
- [11] H. Luo, "Optimization design of cascaded classifiers," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005*, vol. 1. IEEE, 2005, pp. 480–485.

- [12] D. M. Jun and D. L. Jones, "An energy-aware framework for cascaded detection algorithms," in *2010 IEEE Workshop on Signal Processing Systems (SIPS)*. IEEE, 2010, pp. 1–6.
- [13] —, "Cascading signal-model complexity for energy-aware detection," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 65–74, 2013.
- [14] J. Chen, R. Tan, G. Xing, X. Wang, and X. Fu, "Fidelity-aware utilization control for cyber-physical surveillance systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1739–1751, 2012.
- [15] D. Cohen, "Managing resources on a multi-modal sensing device for energy-aware state estimation," Master's thesis, 2013.
- [16] V. C. Raykar, B. Krishnapuram, and S. Yu, "Designing efficient cascaded classifiers: tradeoff between accuracy and cost," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2010, pp. 853–860.
- [17] M. Chen, K. Q. Weinberger, O. Chapelle, D. Kedem, and Z. Xu, "Classifier cascade for minimizing feature evaluation cost," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 218–226.
- [18] B. Emre Ertin, "Polarimetric processing and sequential detection for automatic target recognition systems," Ph.D. dissertation, The Ohio State University, 1999.
- [19] K. Trapeznikov, V. Saligrama, and D. Castañón, "Multi-stage classifier design," *Machine learning*, vol. 92, no. 2-3, pp. 479–502, 2013.
- [20] K. Trapeznikov and V. Saligrama, "Supervised sequential classification under budget constraints," in *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, 2013, pp. 581–589.
- [21] J. Wang, K. Trapeznikov, and V. Saligrama, "An LP for sequential learning under budgets," in *AISTATS*, 2014, pp. 987–995.
- [22] D. M. Jun, L. Le, and D. L. Jones, "Cheap noisy sensors can improve activity monitoring under stringent energy constraints," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 2013, pp. 683–686.
- [23] P. J. Huber, "Robust confidence limits," *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, vol. 10, no. 4, pp. 269–278, 1968.
- [24] —, "Robust statistics," *International Encyclopedia of Statistical Science*, pp. 1248–1251, 2011.
- [25] B. C. Levy, *Principles of signal detection and parameter estimation*. Springer, 2008.
- [26] S. DeBruin, B. Ghena, Y.-S. Kuo, and P. Dutta, "Powerblade: A low-profile, true-power, plug-through energy meter," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15. New York, NY, USA: ACM, 2015. [Online]. Available: <http://doi.acm.org/10.1145/2809695.2809716>
- [27] W. J. Leonard, J. Neal, and R. Ratnam, "Variation of Type B song in the endangered Golden-cheeked Warbler (*Dendroica chrysoparia*)," *The Wilson Journal of Ornithology*, vol. 122, no. 4, pp. 777–780, 2010.
- [28] J.-F. Chamberland and V. V. Veeravalli, "Asymptotic results for decentralized detection in power constrained wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1007–1015, 2004.
- [29] R. D. Smallwood and E. J. Sondik, "The optimal control of partially observable Markov processes over a finite horizon," *Operations Research*, vol. 21, no. 5, pp. 1071–1088, 1973.
- [30] D. P. Bertsekas, "Dynamic programming and stochastic control," 1976.



Long N. Le received the BSEE and MSEE degrees in electrical engineering in 2011 and 2013 from the Ho Chi Minh University of Technology and the University of Illinois at Urbana-Champaign, respectively. During the summer of 2015, he was an intern at the Audio and Acoustics Research Group at Microsoft Research. He was awarded the Best-in-Session by the Semiconductor Research Corporation at TECHCON 2016, in the IoT System Design session. He is currently working toward the Ph.D. degree as a research assistant at the Coordinated Science Laboratory and Beckman Institute of the University of Illinois at Urbana-Champaign. His current research interests include resource-efficient statistical inference and signal processing, with a focus on IoT applications.



Douglas L. Jones received the BSEE, MSEE, and Ph.D. degrees from Rice University, Houston, TX, USA, in 1983, 1986, and 1987, respectively. During the 1987-1988 academic year, he was at the University of Erlangen-Nuremberg, Germany, on a Fulbright postdoctoral fellowship. Since 1988, he has been with the University of Illinois, Urbana-Champaign, IL, USA, where he is currently the director of Advanced Digital Sciences Center (ADSC) and a Professor in Electrical and Computer Engineering, Neuroscience, the Coordinated Science

Laboratory, and the Beckman Institute. He was on sabbatical leave at the University of Washington in Spring 1995 and at the University of California at Berkeley in Spring 2002. In the Spring semester of 1999 he served as the Texas Instruments Visiting Professor at Rice University. His research interests are in digital signal processing and systems, including nonstationary signal analysis, adaptive processing, multisensor data processing, OFDM, and various applications such as low-power implementations, biology and neuroengineering, and advanced hearing aids and other audio systems. He is an author of two DSP laboratory textbooks. Dr. Jones served on the Board of Governors of the IEEE Signal Processing Society from 2002 to 2004. He was selected as the 2003 Connexions Author of the Year.