# Beyond Federated Learning: Survival-Critical Machine Learning

Eric Sturzinger, Mahadev Satyanarayanan
Carnegie Mellon University

***Abstract*—Drawing on parallels with biological immunity, this paper introduces a new use case for learning at the edge called *survival-critical machine learning (SCML).* Unlike federated learning, which assumes supervised learning with pre-labeled data, SCML involves semi-supervised learning in streaming settings where labels may need to be obtained at very low network bandwidth and extreme class imbalance. We show that the recently-developed workflow of *Live Learning* is a good fit for SCML. Starting from a weak bootstrap model, this workflow seamlessly pipelines semi-supervised learning, active learning, and transfer learning, with asynchronous bandwidth-sensitive data transmission for labeling. As improved models evolve at the edge through periodic re-training, the threat detection ability of the SCML system improves. This, in turn, improves the survivability of the host system.**

***Index Terms*—immune systems, biological immunity, live learning, survival, edge computing, mobile computing, machine learning, low bandwidth, computer vision, wireless networks, robotics**

## I. Introduction

The dominant machine learning (ML) paradigm for edge computing today starkinvolves three steps. First, training is done in the cloud with fully-labeled data. Second, the trained deep neural network models (DNNs) are disseminated. Third, the DNNs are used at the edge for inferencing. An exception to this paradigm is *federated learning* [1], a form of supervised learning in which training occurs at the edge rather than in the cloud. The motivation is privacy. By exposing raw data for training only at the point of data capture, information leakage to other nodes and the cloud is greatly reduced. A key requirement of federated learning is that data be already labeled at source, before training begins.

This position paper describes a newly-emerging use case in which learning at the edge is valuable for reasons unrelated to privacy. Federated learning is unsuitable for this new use case because of its assumption of pre-labeled data. Indeed, obtaining labels under conditions of low network bandwidth and extreme class imbalance becomes a major challenge.

In this new use case, called *survival-critical machine learning (SCML),* ML is central to the survival of the system that hosts the learning. In SCML, a misclassification error may lead to destruction of the system. This is in stark contrast to the benign consequences of ML imperfection in well-known use cases today. For example, in an ML-based recommendation system, an erroneous recommendation might result in user annoyance or a lost sale. This is mild compared to system destruction because of misclassification of one threat.

Our confidence in ML has grown to the point where, in the near future, we envision systems in which it is not merely an enhancement to functionality or performance, but is central to survival. This is analogous to biological immune systems, which made their initial appearance on our planet soon after life itself emerged. For example, CRISPR prokaryotic cell defense represents inherited learning in modern bacteria and archaea from virus threats encountered by their dawn-of-life ancestors [2]. Continuous learning to defend against dynamically evolving threats is the essence of an immune system. Although it will take a long time for ML-based threat response systems to attain the sophistication of biological immune systems that have evolved over a billion years, they both rely on continuous learning. SCML is only a speculative vision today, but we predict that its rudiments will appear in real world cyber systems within a decade, if not sooner.

## II. Biological Immunity and SCML

The recent COVID-19 pandemic offers a tutorial on biological immunity. It is estimated that a total of over 700 million people were infected worldwide [3]. Fortunately, this virus was much less lethal than pandemics such as the bubonic plague in the middle ages. Hence, only one percent of those infected (i.e., just over 7 million) died. A disproportionate fraction of the deaths occurred in the first year of the pandemic, prior to the availability of vaccines. While the virus can still be fatal, widespread immunization has made it far less of a threat.

From an ML point of view, immunizing an individual greatly improves his or her body's ability to reliably detect the virus. Reliable threat detection allows a wide arsenal of innate biological defenses to be promptly deployed against the threat. Ideally, this prevents infection. Occasionally, the detection may fail or the defenses may be unsuccessful, and infection may result in spite of immunization. Once infected, there is some low probability that the infected individual will die. The virus continues to mutate, so the threat evolves continuously. In response, new versions of the vaccine are developed for the more recent virus strains. This battle between threat evolution and improved detection will continue far into the future.

The learning in this example is performed by the pharmaceutical companies that create the vaccines. The vaccines thus represent the biological counterparts of pre-trained DNNs in ML. Once a vaccine is received, a human body is able to use the learning embedded in it for improved threat defense. The gap in time between learning and use is small, typically a few

2837-4827/24/$31.00 ©2024 IEEE
DOI 10.1109/SEC62691.2024.00056

weeks to a few months. This mirrors the "train in the cloud, and deploy for inferencing at the edge" paradigm of ML.

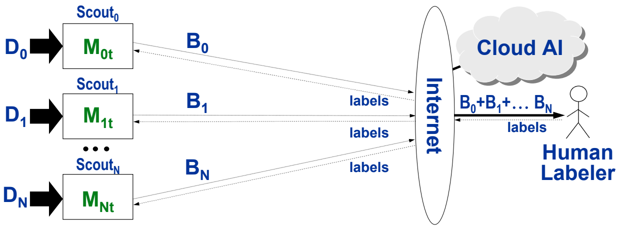
In contrast, the learning embodied by the CRISPR mechanism mentioned earlier (§I) was performed by the ancestors of modern prokaryotic cells. No external entities such as pharmaceutical companies existed to perform "learning in the cloud." That learning was captured as short RNA patterns that are distinctive of a specific virus threat. These distinctive patterns were incorporated into the genetic material of the ancient cells for the benefit of their future descendants. The learning is now valuable across vast stretches of time, on the order of a billion years. Even though modern prokaryotic cells are far less complex than a human, they are able to perform the simple pattern matching needed to use the learned knowedge embodied in CRISPR.

Although we do not have any examples of SCML systems in existence yet, we can use biological immunity to guide us in speculating how such systems could work in the future. The early life forms that developed CRISPR could not rely on any external help. They had to perform unsupervised learning in the face of deadly threats. This is a very slow and error prone process, which must have lead to many fatalities along the way. While the exact evolutionary sequence that led to success is lost in the mists of time, we have proof today that this multi-generational learning process was indeed successful. All of today's organisms that use CRISPR-based defense mechanisms inherit this success.

From an ML viewpoint, we have two advantages that were not available to those early life forms. First, external intelligence is available in the form of humans or cloud-based AI to label data. Second, wireless communication enables that external intelligence to be reached. It also enables progress to be shared with peers to accelerate collective learning. We expect the earliest SCML systems to arise in adversarial settings with rapidly mutating threats. For example, an adversary may use programmable camouflage [4], [5] to morph the visual appearance of a deadly threat and hence evade detection. The adversary will not only try to destroy SCML entities, but also to disrupt communications betweeen them. The need for low observability and jamming resistance may severely constrain wireless network bandwidth [6]. It is likely that the most deadly threats will be those that are rarely encountered. Taken together, these considerations suggest that SCML is likely to involve semi-supervised learning, under conditions of extreme class imbalance and very low network bandwidth. Next, we present an abstract problem formulation with these constraints.

## III. Abstract SCML Problem Statement

The earliest uses of SCML are likely to be for autonomous unmanned vehicles such as aerial drones, satellites, interplanetary spacecraft, and underwater drones. They may occur in law enforcement, military operations and other adversarial settings. We use the term *scout* for an autonomous unmanned vehicle that uses SCML. The critical requirement of a scout is that it has sufficient computing resources and storage to perform training as well as inferencing with on-board resources.



Labeling rate $B_i << D_i$
$D_i$ = data rate into $\text{scout}_i$ $M_{it}$ = model at $\text{scout}_i$ at time $t$

Fig. 1. Abstract Problem Setting of SCML

The term *mission* refers to an extended scout operation with a clear goal and success criterion. The duration of a mission has to be long enough for many threat instances to be encountered, and for the learning that occurs in the early part of the mission to benefit later parts. Since learning is an inherently slow process, short operations cannot benefit from SCML. We expect mission durations to be at least tens of minutes, possibly hours or days. Space missions may last much longer, possibly years or decades.

Without loss of generality, we focus on missions that involve the successful delivery of some valuable payload by a scout from a source to a destination. For example, the payload may be life-saving medical supplies, food, or essential spare parts. A mission involves the risk of scout destruction by threats during its journey. The threats may be bullets, missiles, etc. The scout is protected by an SCML system that continuously tries to detect incoming threats. Once detected, a threat can be neutralized by some *countermeasure (CM)*. Real world CMs may span a wide range. Examples include: projectiles that destroy the threat; chaff, that causes the threat to harmlessly activate; and energy, for evasive maneuvers to escape the threat. Exactly what form CMs take, and how they are deployed is not relevant to SCML. In practice, threat neutralization may not be always successful. Further, there may only be a limited number of CMs available on the scout. Once these CMs are exhausted, the system is totally vulnerable to all further incoming threats even if they are detected.

Some missions may involve a team of scouts. This may be necessary to deliver a large enough payload, or to provide redundancy in case of destruction of some scouts. A team-based mission leads to the question of how to define success when some scouts die but others survive. The simplest criterion is to require all scouts to reach their destination. A less stringent criterion is to require only $M$ out of $N$ scouts to arrive.

Figure 1 shows a team of SCML-protected scouts on a mission. They encounter instances of a new deadly threat, for which no accurate DNN detector yet exists. The mismatch between incoming sensor data rate at scouts and outgoing wireless backhaul bandwidth may be severe. There may also be extreme class imbalance in the incoming data because only a few threats of the novel class are present amidst the vast amount of background data observed. The trifecta of extremes (very low network bandwidth, extreme class imbalance, and target novelty) makes this a challenging ML problem setting.

To support semi-supervised learning, the cloud and a human labeler are reachable over a low bandwidth network link in Figure 1. Unfortunately, the severe bandwidth mismatch means that only a tiny fraction of the data can be labeled. Random sampling to reduce bandwidth demand is unlikely to be useful. Because of the extreme class imbalance, it will result in most labeled data being easy negatives. Very few true positives (TPs) are likely to be labeled. For learning to be effective, the proportion of TPs in a training set needs to be much higher than their rarity in the wild. This suggests the need for intelligent data selection for transmission. We describe how to do this in the next section.

## IV. Live Learning for SCML

Extreme low network bandwidth makes transmission of newly-trained models from the cloud to scouts painfully slow. A modern DNN is easily many tens of MB in size (Figure 2). At bandwidths of a few kbps, the transmission would take on the order of hours. Model compression techniques such as DeepIoT [8] report size reductions of 90–99% on sequential model architectures such as LeNet and VGG. Unfortunately, as Figure 2 shows, DeepIoT is less effective on more modern DNNs. Training on scouts avoids model transmission delays. The training time may be longer because of less powerful hardware, but this is more than compensated by completely avoiding model transmission [7].

| | Size (MB) | Compression Ratio |
|---|---|---|
| ResNet-50 | 97.69 | 2.53 |
| YOLOv5-small | 27.60 | 2.31 |
| EfficientNet-B4 | 74.26 | 2.68 |

Source: George et al [7]

Fig. 2. DeepIoT Compression

The scarcity of the new threat class poses a conundrum. On the one hand, extremely rare threats lower the risk of destruction. However, it also implies slow learning because so few TPs can be gathered for a training set. Frequent exposure to threats (and survival by good luck) leads to much faster learning and model improvement. Learning can be accelerated by sharing new knowledge between scouts. This has no parallel in biological immunity, except through inherited knowledge such as CRISPR and the sharing of vaccines. Sharing knowledge via a wireless network benefits current scouts, not just future generations. What to share thus becomes a relevant question. In the face of extreme class imbalance, it is always worth sharing TPs after labeling. With more bandwidth, it may be worth sharing all labeled data. This is still only a small fraction of total incoming data.

Relative to the new threat class, a scout thus faces a chicken-or-egg problem. The only way to improve its threat detection ability is to train a new model on the scout. This requires a large enough training set, with a disproportionately high number of TPs relative to their naturally-occurring base rate. How does a scout obtain a large number of labeled TPs, when it does not yet possess an accurate DNN to select them for transmission to a labeler?

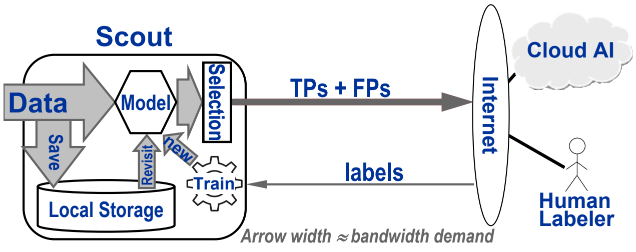


Fig. 3. End-to-End Pipeline of Hawk for SCML

*Live Learning* offers a possible solution. This pipelined and iterative ML workflow collects training sets for new, rare phenomena in bandwidth-challenged settings [7]. We repurpose Live Learning for SCML. An open source implementation of Live Learning is embodied in a system called *Hawk* [9]. Starting from a weak model that is trained via few-shot learning (FSL) on just a few initial examples, Hawk seamlessly pipelines semi-supervised learning, active learning, and transfer learning, with asynchronous bandwidth-sensitive data transmission for labeling. When a significant number of TPs have been labeled, Hawk trains an improved model to replace its old model. This iterative workflow is Live Learning, which continues until sufficient TPs have been collected. The workflow is independent of the specific DNN architecture.

Figure 3 illustrates the Hawk processing pipeline. Fresh data arrives continuously at a high rate into a scout. In this paper, we assume that the sensing modality is visible light, resulting in a video stream. However, Live Learning and SCML could work equally well with other sensing modalities such as radar or sonar. Incoming data is both written to local storage, and processed. The processing consists of three steps. The video stream is first decoded into individual frames. Each frame is broken up into small tiles, and the tile stream is inferenced using the current model. After inferencing, a small subset of the tiles is selected for transmission and labeling. A tunable data sourcing policy guides processing new incoming data versus re-processing previously discarded data. Transmitted items are labeled by a human expert or cloud AI. Labels are transmitted back to the scout as they are generated. With multiple scouts, precious TPs are shared across the team so that models improve at their cumulative TP discovery rate. Pseudo-labeled negatives are obtained locally on each scout.

For simplicity, we focus on the ML problem of classification. In practice, an SCML system also needs to localize the objects — i.e. the underlying problem is object detection in live data, not just classification. Since that further level of complexity adds no new insights to our discussion, we focus on classification. For every incoming data item, a scout asks and answers: *"Is there a threat at this moment?"* It uses the currently installed model to answer this question. Over time, its answers become more accurate because of model improvement. If the scout survives the early part of its mission, its ability to detect the new threat class continues to improve over the duration of the mission. Faster learning is valuable because it improves the chances of survival.

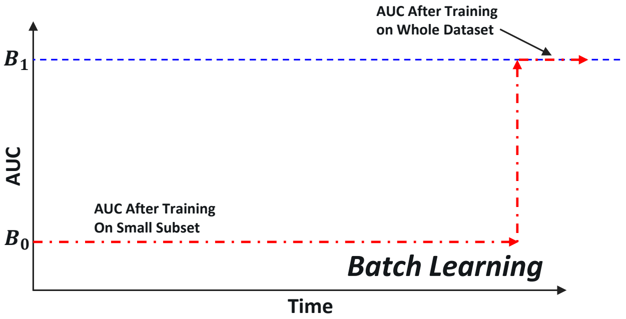


Fig. 4. AUC Improvement with Batch Learning

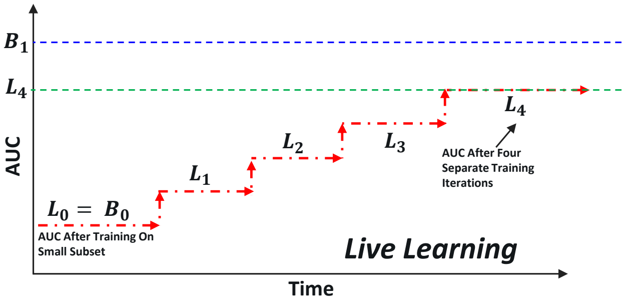


Fig. 5. AUC Improvement with Live Learning

## V. The Role of Time

Historically, time has played no role in characterizing the accuracy of a classifier. The well-known metrics of precision and recall are all that matter. The time needed to create and use a model (i.e., data collection, labeling, training, and inferencing) is not relevant. These are viewed as secondary metrics, typically outside scope in discussions of accuracy. In SCML, time can no longer be relegated to a secondary role. A single classification error can lead to an undetected threat destroying the scout. There will be no further opportunity for that model to redeem itself through future successes. In SCML, an early mistake means that the system dies young. Early survival, on the other hand, leads to later mistakes being less likely because of model improvement. Thus, time has to be viewed on an equal footing with precision and recall.

The conceptual role of time is captured by Figure 4 and Figure 5, which contrast Batch Learning (classic ML) and Live Learning. The Y axis is the area under the precision-recall curve (AUC), which is a widely-used metric for the accuracy of a classifier. In Batch Learning (Figure 4), training is deferred until ample data has been collected and labeled. In Batch Learning, the AUC is not improved for the current mission during data collection, labeling, training set construction, and training. The fruits of this work are available only to future missions. In contrast, Live Learning improves AUC during the current mission. Although it may be unable to achieve the same optimal model as Batch Learning (because of the constraints of online rather than offline learning), it improves survivability for the current mission. This improvement is correlated with the increase in area under the AUC curve (i.e., integral of AUC over mission duration).

With Batch Learning, missions are not even attempted until an accurate model exists. Unfortunately, SCML has to cope with a world in which threats are not static, but evolve. Critical missions may have to be attempted even without a high-quality model for a new threat class. For that class, a weak model that is based on a small training set may be all that is available at the start of a mission. During the mission, Live Learning can be used on incoming data to improve the model. More threats will be encountered on longer missions. Hence, there is greater danger that a threat will get through undetected and destroy the scout. However, if the scout survives the early part of a long mission, Live Learning has significant opportunity to improve the model. There is thus a delicate balance between survival and learning, with time at its heart (Figure 5).

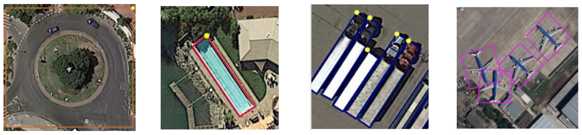
The key advantage of Batch Learning is that all data is available up front for analysis to create an optimal training set. This can yield the best model achievable from that training data for a given model architecture. Unfortunately, this optimality is of little value in an SCML setting — if the mission is attempted, the scout is unlikely to survive long enough to benefit. Like the proverbial tortoise relative to a hare, Live Learning settles for much smaller incremental gains. However, these gains are available during the current mission. The gains are suboptimal relative to Batch Learning because they have to be made in an online setting. As Figure 5 illustrates, the result is a sub-optimal final AUC ($L_4 < B_1$) but it is reached in incremental steps.

## VI. Preliminary Results

Using the Hawk prototype, we conduct an initial exploration of the value of Live Learning for improving survivability in SCML. Since SCML systems do not exist yet, there are no relevant SCML datasets to use in the evaluation. We therefore re-purpose a dataset from a non-SCML setting. We arbitrarily designate one of its classes as a deadly threat, while other classes are benign. From an ML point of view, no compromise arises from this evaluation strategy. CMs remain abstract in our investigation, and are not related to the dataset. As mentioned earlier (§III), real world CMs may involve projectiles, chaff, energy for evasive maneuvers, etc.

### A. Experimental Setup

Each scout consists of a 6-core 3.6 GHz Intel® Xeon® E5–1650v4 processor, 32 GB memory, 4 TB disk storage for image data, and an NVIDIA® GTX 1060 GPU. Today, a typical scout would be configured with embedded hardware such as an NVIDIA Jetson AGX Orin [10], weighing 1.58 kg. Such hardware could easily be carried by an autonomous drone, unmanned spacecraft, or underwater immersible. We assume a network backhaul bandwidth of 50 Kbps to the cloud and human labeler, and a bandwidth of 1 Mbps between scouts. The 50 Kbps value reflects the maximum data rate of a LoRaWAN connection [11]. Our experiments use a team of 7 scouts, each configured as above. The results of 3 runs of each experiment, varying random number seeds, are reported.



(a) Roundabout (TPs=336) (b) Swimming Pool (TPs=335) (c) Large Vehicle (TPs=357) (d) Airplane (TPs=350)

Fig. 6. Examples of DOTA Target Classes

| | Class: Roundabout | | Class: Pool | |
|---|---|---|---|---|
| Gen | Time(s) | AUC | Time(s) | AUC |
| 0 | 0 | 0.17 (0) | 0 | .002 (0) |
| 1 | 151 (8) | 0.33 (0.08) | 894 (152) | 0.27 (0.06) |
| 2 | 305 (39) | 0.34 (0.07) | 1061 (113) | 0.40 (0.07) |
| 3 | 491 (119) | 0.44 (0.07) | 1241 (99) | 0.56 (0.04) |
| 4 | 700 (74) | 0.48 (0.06) | 1478 (62) | 0.67 (0.03) |
| 5 | 951 (53) | 0.53 (0.05) | 1767 (115) | 0.72 (0.01) |
| 6 | 1412 (2) | 0.56 (0.07) | 2102 (158) | 0.78 (0.02) |
| 7 | 2023 (101) | 0.60 (0.06) | 2535 (109) | 0.80 (0.01) |
| 8 | 2660 (163) | 0.62 (0.02) | 3149 (110) | 0.82 (0.01) |
| | 3605 (3) | | 3590 (22) | |

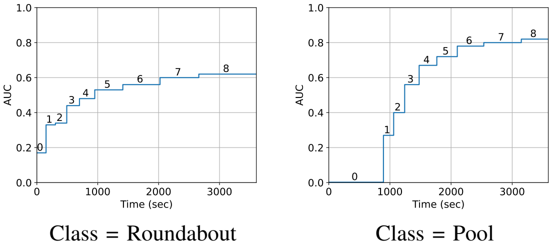
Figures in parentheses are standard deviations across 3 runs.

Fig. 7. Mission Timeline for Live Learning

Our experiments use a publicly-available 15-class drone surveillance dataset called DOTA v1.0 [12]. Examples from four of the 15 classes are shown in Figure 6. The curated DOTA-derived dataset that we use in our experiments has a threat class base rate on the order of 0.1%. In other words, approximately one in 1000 samples is a TP. This gives the class-specific TP counts shown in Figure 6. There are 2806 images, half of which are used as mission data. Image resolution ranges from $800 \times 800$ pixels to $4000 \times 4000$ pixels. Images are tiled into $256 \times 256$ pixels, yielding 252,231 usable tiles. During a mission, a scout receives an average of 200 tiles in 20 seconds. With the data striped across 7 scouts, this gives a total mission duration of roughly 60 minutes.

### B. Live Learning Timeline

The timeline of Live Learning on a scout is dependent on many factors, including dataset attributes, threat class attributes (especially the class imbalance), and many ML hyperparameters. For our specific experimental setup (§VI-A), Figure 7 presents the observed Live Learning timeline for two representative threat classes. For the class "Roundabout", the second and third columns of Figure 7(a) show this timeline. The initial model, based on a bootstrap training set containing just 20 TPs, is weak. Its AUC (area under the precision-recall curve on a held-out test set) is merely 0.17. However, in the face of our extreme class imbalance, even 0.17 is much better than random classification. The first new model is installed 151 seconds into the mission, and has a significantly improved AUC of 0.33. At 305 seconds, a slightly better model with AUC 0.34 is installed; at 491 seconds, a model with AUC 0.44 is installed; and so on. The AUC improvement is mostly monotonic, with occasional plateaus or reversals due to the randomness inherent in the learning process. The final model with AUC 0.62 has improved far beyond the initial model.



Class = Roundabout Class = Pool

Fig. 8. Visualization of Live Learning Timeline

| Class | Tiles Processed | Tiles Labeled | TPs Found | TPs Total |
|---|---|---|---|---|
| Roundabout | 252,231 (0) | 3780 (0) | 250 (9) | 336 |
| Pool | 252,231 (0) | 3780 (0) | 236 (11) | 335 |

These results are from the same experiments as Figure 7. Numbers in parentheses are standard deviations. "TPs Total" refers to the number of TPs in ground truth.

Fig. 9. Transmission and Labeling Efficiency of Live Learning

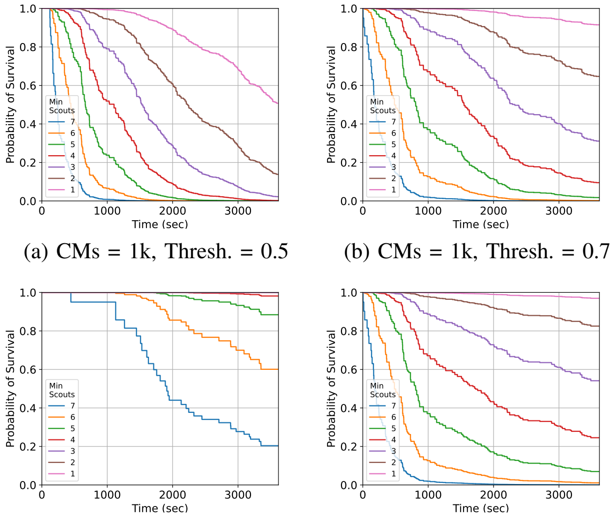
The timeline of Live Learning is even more dramatic for the class "Pool", as shown by the fourth and firth columns of Figure 7. The initial model has an AUC of merely 0.002, but it improves in 8 steps to 0.82 by the end of the mission.

Using the canonical Live Learning timeline shown earlier in Figure 5, Figure 8 presents a visualization of the data in Figure 7. For both classes "Roundabout" and "Pool", significant improvement in model accuracy is seen over the life of the mission. While the magnitude of the improvement is class dependent, our results confirm that the improvement is non-trivial for both classes.

Figure 9 shows the transmission and labeling efficiency of Live Learning for the experiments shown in Figure 7. For both classes, the total number of tiles processed by the scouts is 252,231. Of these, only 3780 were transmitted for labeling by the human in the loop. In spite of the extreme class imbalance (~0.1% base rate), 250 TPs were found out of 336 ground truth TPs for class Roundabout, and 236 TPs were found out of 335 ground truth TPs for class Pool. This confirms that Live Learning is frugal in its use of network bandwidth and human labeling effort, but highly effective in discovering TPs.

### C. Benefits of Live Learning for Survivability

From an SCML viewpoint, the key question is whether improved learning leads to improved survivability. Assuming that a CM is 100% effective in neutralizing a threat, there are two ways in which a scout can die. First, a TP may be incorrectly classified, thus resulting in a false negative (FN). Second, in spite of correct threat classification, there may be no CMs left. This may arise because too few CMs were provisioned for the mission. Or, it may arise because too many CMs were used earlier in the mission because of too many false positives (FPs). Too low a classification threshold increases the chances of this eventuality. In either case, the fate of the scout is then determined solely by the *lethality* of the threat, which is expressed as a probability of destruction ($\gamma$). The survival of the scout then depends on pure luck.



(a) CMs = 1k, Thresh. = 0.5

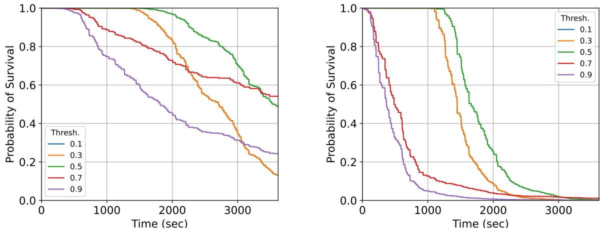
(b) CMs = 1k, Thresh. = 0.7

(c) CMs = 50k, Thresh. = 0.5

(d) CMs = 50k, Thresh. = 0.7

These graphs show the probability of survival of at least M of N scouts for 5% lethality with the given thresholds and numbers of countermeasures.

Fig. 10. Survivability of M of N Scouts by CMs and Threshold



(a) M = 3 Scouts

(b) M = 6 Scouts

These graphs show the probability of survival at respective thresholds for 5% lethality and 10k CMs for the given numbers of M scouts.

Fig. 11. Survivability of M of N Scouts by Threshold

Suppose no less than $M$ out of $N$ scouts must reach the destination for a mission of duration $T$ to be successful. The likelihood of mission success is the probability that at least $M$ of $N$ scouts are still alive at time $T$. Because learning may vary across scouts, the binomial distribution does not apply. However, the Poisson Binomial Distribution [13] is applicable. Using $C(x, y)$ to enumerate all the possible ways of choosing $y$ things from $x$, Equation 1 gives the probability that exactly $M$ scouts are alive at time $T$. When $M = N$, all scouts need to reach the destination. For the more relaxed criterion of $M < N$, Equation 2 gives the probability of mission success where at least $M$ scouts are alive at time $T$.

$$P_T(s = M) = \sum_{I \in C(N,M)} \prod_{i \in I} p_T^i \prod_{j \notin I} (1 - p_T^j) \tag{1}$$

$$P_T(s \geq M) = \sum_{k=M}^{N} \sum_{I \in C(N,k)} \prod_{i \in I} p_T^i \prod_{j \notin I} (1 - p_T^j) \tag{2}$$

Using these equations and the learning timeline shown in Figure 8, the likelihood of mission success is shown in Figure 10 as a function of classifier threshold, number of CMs, and the number of scouts that need to reach the destination. In all cases, a threat lethality of 5% is used. With a small number of CMs and a threshold of 0.5, only 1 or 2 scouts have any possibility of surviving the mission, as shown in Figure 10(a). With a slightly higher threshold of 0.7, as in Figure 10(b), up to 3 scouts have a reasonable chance of surviving. This is because a higher threshold decreases the rate of CM depletion, increasing survivability. Figures 10(c) and (d) show survivability curves at thresholds 0.5 and 0.7 for 50k CMs. With many more CMs, we are unlikely to run out; hence, FNs are the only source of danger. For example, the probability of survival of at least 6 scouts is roughly 60% in Figure 10(c), while it is nearly 0% in Figure 10(d) because of many more FNs at a threshold of 0.7.

Our analysis also provides us with planning options to determine the optimal threshold at which to operate the SCML system. Figure 11 shows the impact of threshold on survivability, with a lethality of 5% and 10,000 CMs per scout. If the mission duration were 2000 seconds, the orange curve representing a threshold of 0.3 in Figure 11(a) would be a better choice than 0.9 (purple curve). However, for missions longer than 3000 seconds, a threshold of 0.9 becomes optimal. Requiring 6 scouts to survive clearly lowers all probabilities in Figure 11(b). Many similar insights spanning survivability as a function of the number of CMs, threshold, and success criteria can be obtained from Figure 11.

## VII. Future Work

Our preliminary evaluation of Live Learning (§VI) indicates that it does indeed improve survivability for a mission, and is therefore a viable approach to implementing SCML. These early results suggest that deeper and broader effort in this space would be worthwhile. We discuss below some of the relevant topics that could be part of such a research agenda.

**Broader and More Realistic Validation:** Our experiments in this paper use a single dataset (DOTA v1.0) and two threat classes from it (Roundabout and Pool). A more complete evaluation, including coverage of all 15 classes of DOTA as well as datasets from planetary exploration (HiRISE [14]) and underwater sensing (Brackish [15]), can be found in a recent paper [16]. That work also presents an analytical model for SCML, and the results of experimental tradeoffs in CM use and learning bias. However, extrapolating from these non-SCML datasets may be of limited validity. Real-world threat distributions may not match these extrapolations. Obtaining threat profiles from real-world applications of SCML and using them in Live Learning experiments would therefore be valuable. Since such data can only be obtained from working SCML systems, data collection has to be deferred until their appearace. In the interm, it may be valuable to explore datasets from adversarial settings.

**Multi-generational SCML:** This work assumes that when multiple scouts are available, they are launched together as a team and that each has the same bootstrap model. In contrast, if the launch of individual scouts is temporally staggered, the later scouts can benefit from the learning of the earlier scouts. Of course, the rate of that learning will be slower because fewer threats are encountered by the smaller initial cohort of scouts. How these factors balance out would be valuable to study. A variant of this approach is to maintain a fixed number of scouts in flight. When a scout dies, a new one is launched to replace it. That scout can start with the best model learned so far. To a certain extent, this mirrors real life where the survival of a species is not critically dependent on the survival of a single individual. Rather, each generation has a finite lifetime but its learning can be inherited by the next generation and thereby give it a head start. Of course, the risk is that old learning is obsolete and possibly counter-productive. It would be valuable to explore these concepts and their associated tradeoffs.

**Biological Immunity:** This paper has drawn heavily upon parallels to biological immunity to makes its case for SCML. Coming full circle, it would be valuable to model the immune systems of organisms from the viewpoint of SCML and Live Learning. Humans are highly evolved, and likely to be too complex for such evaluation. However, modeling the early evolutionary steps of biological immunity could be valuable. Such an investigation could be especially valuable in identifying a plausible evolutionary sequence that could have led to the emergence of a mechanism like CRISPR. Looking further ahead, one can ask whether an SCML system can be designed to augment an organism's own biological immunity. Such augmentation is likely to be decades away, but it is not too early to begin speculation about how to achieve it.

## VIII. Conclusion

In closing, this paper has introduced a new use case for learning at the edge called SCML. Unlike federated learning, which assumes supervised learning with pre-labeled data, SCML involves semi-supervised learning in streaming settings where labels may need to be obtained at very low network bandwidth and extreme class imbalance. We have shown that the recently-developed workflow of Live Learning is a good fit for SCML. Starting from a weak bootstrap model, this workflow seamlessly pipelines semi-supervised learning, active learning, and transfer learning, with asynchronous bandwidth-sensitive data transmission for labeling. As improved models evolve through periodic re-training, the threat detection ability of the SCML system improves. This, in turn, improves the survivability of the host system.

## Acknowledgments

We thank the anonymous reviewers for their thoughtful and constructive feedback that improved the presentation of this paper. This material is based upon work supported by the U.S. Army Research Office and the U.S. Army Futures Command under Contract No. W519TC-23-C-0003, by the United States Navy under award number N00174-23-1-0001, and by the National Science Foundation under grant number CNS-2106862. The content of the information does not necessarily reflect the position or the policy of the government and no official endorsement should be inferred. This work was done in the CMU Living Edge Lab, which is supported by Intel, Arm, Vodafone, Deutsche Telekom, CableLabs, Crown Castle, InterDigital, Seagate, Microsoft, the VMware University Research Fund, IAI, and the Conklin Kistler family fund. Any opinions, findings, conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the view(s) of their employers or the above funding sources.

## References

[1] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, May 2020.

[2] E. V. Koonin and K. S. Makarova, "Origins and evolution of CRISPR-Cas systems," *Philosophical Transactions of the Royal Society B*, vol. 374, no. 20180087, 2019, (http://dx.doi.org/10.1098/rstb.2018.0087).

[3] "COVID-19 Coronavirus Pandemic," (https://www.worldometers.info/coronavirus/), Last accessed on September 15 2024.

[4] "Better Camouflage is Needed to Hide from New Electronic Sensors," *The Economist*, March 29, 2023.

[5] C. Feng, M. Mao, X. Zhang, Y. Liao, X. Xiao, H. Liu, and K. Liu, "Programmable Microfluidics for Dynamic Multiband Camouflage," *Microsystems & Nanoengineering*, vol. 9, no. 1, April 2023, (https://doi.org/10.1038/s41378-023-00494-3).

[6] D. Adamy, *EW 103: Tactical Battlefield Communications Electronic Warfare*. Artech House, 2008.

[7] S. George, H. Turki, Z. Feng, D. Ramanan, P. Pillai, and M. Satyanarayanan, "Low-Bandwidth Self-Improving Transmission of Rare Training Data," in *The 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom 23)*, Madrid, Spain, October 2023.

[8] S. Yao, Y. Zhao, A. Zhang, L. Su, and T. Abdelzaher, "DeepIoT: Compressing Deep Neural Network Structures for Sensing Systems with a Compressor-Critic Framework," in *Proceedings of SenSys '17*, Delft, Netherlands, 2017.

[9] "Hawk Source Code," (https://github.com/cmusatyalab/hawk), last accessed September 21, 2024.

[10] "NVIDIA Jetson Orin," (https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/), last accessed July 22, 2024.

[11] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the Limits of LoRaWAN," *IEEE Communications magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[12] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A Large-Scale Dataset for Object Detection in Aerial Images," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[13] Y. H. Wang, "On the number of successes in independent trials," *Statistica Sinica*, pp. 295–312, 1993.

[14] G. Doran, S. Lu, L. Mandrake, and K. Wagstaff, "Mars orbital image (hirise) labeled data set version 3," *NASA: Washington, DC, USA*, 2019.

[15] M. Pedersen, Bruslund, J. Haurum, R. Gade, and T. Moeslund, "Detection of marine animals in a new underwater dataset with varying visibility," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[16] E. Sturzinger, J. Harkes, P. Pillai, and M. Satyanarayanan, "Edge-based Live Learning for Robot Survival," *IEEE Transactions on Emerging Topics in Computing*, 2024, (https://ieeexplore.ieee.org/document/10721342).