
# Protecting Adaptive Sampling from Information Leakage on Low-Power Sensors

Tejas Kannan
University of Chicago
Chicago, IL, USA
tkannan@uchicago.edu

Henry Hoffmann
University of Chicago
Chicago, IL, USA
hankhoffmann@cs.uchicago.edu

## ABSTRACT

Adaptive sampling is a powerful family of algorithms for managing energy consumption on low-power sensors. These algorithms use captured measurements to control the sensor's collection rate, leading to near-optimal error under energy constraints. Adaptive sampling's data-driven nature, however, comes at a cost in privacy. In this work, we demonstrate how the collection rates of general adaptive policies leak information about captured measurements. Further, individual adaptive policies display this leakage on multiple tasks. This result presents a challenge in maintaining privacy for sensors using energy-efficient batched communication. In this context, the size of measurement batches exposes the sampling policy's collection rate. Thus, an attacker who monitors the encrypted link between sensor and server can use message lengths to uncover information about the captured values. We address this side-channel by introducing a framework called Adaptive Group Encoding (AGE) that protects any periodic adaptive sampler. AGE uses quantization to encode all batches as fixed-length messages, making message sizes independent of the collection rate. AGE reduces the quantization error through a series of transformations. The proposed framework preserves the low error of adaptive sampling while preventing information leakage and incurring negligible energy overhead.

## CCS CONCEPTS

• **Computer systems organization → Embedded systems**.

## KEYWORDS

Adaptive Sampling, Embedded Systems, Data Privacy, Lossy Data Encoding

**ACM Reference Format:**
Tejas Kannan and Henry Hoffmann. 2022. Protecting Adaptive Sampling from Information Leakage on Low-Power Sensors. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '22), February 28 – March 4, 2022, Lausanne, Switzerland*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3503222.3507775


*ASPLOS '22, February 28 – March 4, 2022, Lausanne, Switzerland*

ACM ISBN 978-1-4503-9205-1/22/02…$15.00
https://doi.org/10.1145/3503222.3507775

## 1 INTRODUCTION

Low-power sensors have applications in environmental monitoring [70, 111] and healthcare [54, 92]. Sensors monitor their surroundings by collecting measurements and transmitting values to a server for analysis. In many applications, sensors lack access to continuous power [31, 111, 119]; such nodes instead use batteries or harvested energy. Thus, energy is a critical resource, and operators desire sensors with long lifetimes [119]. Subsampling is one strategy to elongate sensor lifetime [63]. Rather than capturing and communicating all measurements, individual nodes collect and transmit a subset of values; a statistical model is used to infer the skipped elements [25, 32, 43]. This design uses the insight that both sensing and communication incur a high energy cost [6, 31, 43, 46].

Subsampling trades energy for error. When sensors collect fewer elements, the server must infer more values. Adaptive sampling is one strategy that navigates this tradeoff in a near-optimal manner. For a given energy constraint, these algorithms sample frequently in unpredictable environments and compensate by collecting fewer values in predictable settings. For example, consider a smartwatch that performs activity detection with an accelerometer. When the wearer is sitting, the acceleration measurements remain constant; when a person is running, the values change rapidly [29]. Thus, the policy samples infrequently during sitting events and spends the excess energy during less-predictable running events. The ability to allocate energy based on the observed data allows adaptive algorithms to exhibit low error while meeting energy constraints, making it an attractive option for low-power devices [43, 115].

As embedded sensing grows in popularity, maintaining data privacy is increasingly important [10, 28, 109]. From this perspective, adaptive sampling faces a significant challenge [102]. By design, adaptive policies display different collection rates in different environments, thus linking their collection pattern to the captured data. This property introduces a possible side-channel: if sensors expose their collection rates within their (encrypted) communication patterns, they may leak information about the captured values. Indeed, previous work shows how attackers can link communication patterns to sensor values for specific IoT devices [12, 16, 29, 101].

Existing work [12, 101] that exploits the communication patterns of adaptive behavior targets IoT devices that vary their transmission times. For example, the FATS attack [101] uses the time between messages to uncover activities in a smart home. One suggested defense is to employ *periodic transmissions* [101] which eliminate the variance in message times. This defense increases device latency by delaying transmissions and sending values in batches. This latency is manageable as low-power sensors already employ batching to conserve energy [104, 119]; batched communication minimizes the time that radio modules spend in active mode [73, 74].

In this work, we demonstrate how periodic transmissions do not protect general-purpose adaptive sampling policies. By "general-purpose," we refer to policies at use widely-applicable statistical models (e.g. linear comparisons [25, 96]) instead of narrow, task-specific knowledge. As adaptive policies vary their collection rates, they create batched messages with a size proportional to the number of collected values. This property holds even when sensors use encryption. Across multiple tasks, we show how three general-purpose adaptive techniques [22, 25, 96] exhibit a distinct relationship between message sizes and sensed events. Thus, message sizes yield an exploitable side-channel for adaptive policies. In the worse case, an attacker can infer over 94% of events using encrypted message sizes. This worst-case leakage occurs on tasks involving sensitive data. For example, an attacker can use message sizes to infer the occurrence of an epileptic seizure [112] with 100% accuracy (§5.4). This issue prevents sensors that require data privacy from realizing the benefits of adaptive sampling.

A solution to close this side-channel is to send fixed-length messages, breaking the link between message size and collection rate. The conventional way to enforce this consistency is to pad messages to the largest possible batch size [20, 36]. Padding hides the true data length by adding meaningless bits and increasing the communication volume. This strategy is impractical for low-power sensors due to the high energy cost of communication [43, 74]. Sensors require a defense with a negligible energy cost: if the strategy increases the energy consumption, it counteracts the benefits of adaptive sampling.

To address this problem, we propose a framework called Adaptive Group Encoding (AGE) that protects adaptive sampling under the constraints of low-power sensors. Unlike traffic padding, AGE produces fixed-length messages smaller than the standard adaptive policy's average message size. AGE is a general defense that works with multiple adaptive policies, sensing tasks, and encryption algorithms. A key challenge to achieving this generality is that AGE cannot make strong assumptions about the policy's behavior or the sensor's data values. AGE ensures same-sized messages using a general lossy technique based on fixed-point quantization. The procedure minimizes its encoding error by supplementing standard quantization with three transformations. AGE first prunes measurements to handle cases where the policy exhibits extreme over-sampling. Second, AGE adapts to data ranges with low space overhead by applying run-length encoding (RLE) to data exponents. This step groups values with the same exponent. AGE handles cases where RLE delivers poor compression by merging similar exponent groups. Finally, AGE sets the bit width for each group and quantizes measurements to this width. The algorithm selects these widths to ensure the result meets the target length. This group-based strategy allows AGE to functionally mimic fractional bit widths, leading to better utilization of the available message space.

Across nine sensing tasks and three adaptive policies, AGE incurs roughly 1% higher error than standard adaptive sampling. This additional error is manageable, as adaptive sampling with AGE obtains over 11% lower error than uniform sampling. AGE's low error *does not* compromise security. AGE protects adaptive policies from all information leakage through message sizes, and communicating with AGE prevents an attacker from doing any better than predicting the most frequent event. Further, AGE's low-overhead design outperforms prior solutions based on message padding. Under equivalent energy constraints, AGE obtains lower error than padding on over 94% of budgets. This result stems from budget violations caused by the overhead of padding. Finally, AGE consistently performs better than quantization alone, achieving lower error on over 98% of constraints. In summary, AGE successfully enables adaptive policies to obtain low error without suffering from information leakage or energy budget violations.

We make the following contributions in this work.

(1) We demonstrate how general-purpose adaptive sampling policies leak information through their collection patterns on multiple distinct tasks. This leakage occurs even though these policies do not use task-specific designs.
(2) We present a practical attack against adaptive sampling algorithms for sensors that use energy-efficient batched communication. In the worst case, this attack can use message lengths to infer over 94% of sensed events.
(3) We introduce a novel framework called Adaptive Group Encoding (AGE). This system uses lossy encoding to protect any adaptive sampling algorithm from leaking information through batched message sizes on low-power devices. AGE closes this side-channel with negligible energy overhead[1].

## 2 BACKGROUND AND SYSTEM MODEL

This work protects adaptive sampling algorithms on low-power devices from leaking information through their communication patterns. This section provides background on both low-power sensors (§2.1) and adaptive sampling methods (§2.2).

### 2.1 Low-Power Sensors and System Model

We consider a standard sensing model composed of battery-powered sensors and a centralized server [46, 111]. Sensor nodes consist of a microcontroller unit (MCU) equipped with sensing hardware. Each device captures measurements and transmits the readings to the server over an encrypted wireless link. We consider sensors that communicate at regular intervals. This design matches a common approach in low-power sensing: batched communication. Batching data into a single communication event saves energy by both maximizing the time spent in sleep mode and amortizing the radio start-up costs over multiple measurements [73, 74, 104]. Periodic communication increases the latency for offloading readings, but this overhead is offset by the over 4× lower energy when batching [74]. Indeed, systems such as FarmBeats [111] and ZebraNet [119] employ sensors with periodic, batched communication. Furthermore, this latency becomes less important when detecting events from sequences of measurements. In this framework, the server requires all sequential elements to perform inference; thus, transmitting batches does not hinder the event detection pipeline. As a concrete example, consider the task of activity recognition with wearable devices [8]. Each battery-powered wearable captures accelerometer and gyroscope measurements over time. The sensor transmits measurements to a server, which uses sequences of readings to detect activities such as running or walking.

[1] All code is available at https://github.com/tejaskannan/adaptive-group-encoding.

The finite lifetime of battery-powered sensors complicates the process of recording and communicating measurements. This limited power makes energy conservation a priority, as frequent maintenance is costly and undesirable. Finite-capacity energy sources lead to long-term *energy budgets* where sensors must use their residual battery capacity to meet an operator-defined uptime. For example, ZebraNet sensors are designed to last for at least 72 hours on battery power alone [119].

Sensors consume energy through three tasks: collection, processing, and communication. Of these tasks, collection and communication consume a majority of energy [6, 32, 43]. For example, a commodity HM-10 Bluetooth Low Energy (BLE) radio consumes about 25mJ to connect and send a 40-byte message. In contrast, a TI MSP430 FR5994 MCU requires roughly 0.4mW per clock MHz [3]; this figure is an order of magnitude less than wireless communication. The high energy cost of radio modules means that embedded applications strive to limit their use of wireless communication. On low-power devices, it is prohibitive to design systems that increase the amount of communication.

## 2.2 Subsampling and Adaptive Behavior

Subsampling allows sensors to manage their energy consumption [5, 7, 25, 96] by collecting and sending a subset of values. The server receives this subset and interpolates the full sequence [63]. A sampling policy's energy is proportional to its collection rate. Thus, sampling comes with a tradeoff in error; when sensors capture fewer values, the server must infer more.

Adaptive sampling navigates this tradeoff in a near-optimal manner. These procedures use observed data to determine when to collect the next element. Adaptive policies will capture more measurements on sequences with high variance and fewer samples during low-variance periods [63]. For example, Figure 1 shows sampling two sequences of 25 acceleration values with a budget corresponding to a 70% average collection rate [112]. A random sampler will capture 17 samples per sequence. This rigid strategy is suboptimal, and an adaptive policy [25] can better allocate the budget. Across both sequences, the adaptive policy achieves over 2.9× lower error by under-sampling sequence one and over-sampling sequence two. This lower error occurs even though the adaptive sampler captures two fewer values than the random policy.

Under energy constraints, this data-dependent behavior allows adaptive policies to achieve lower error than static sampling. The benefits, however, come at a cost in privacy. For the adaptive policy, collecting either 10 or 22 elements (from the top or bottom of Figure 1) leaks whether the event is walking or running, respectively. Thus, an attacker who observes the collection count can infer the sensed event. The random policy does not suffer from this problem because its rate is fixed and independent of the underlying data. From this discussion, we highlight two important aspects of the system model.

(1) Long-term energy budgets allow for variance in the sampling policy. The sampler can change its energy per sequence as long as the final energy meets the budget.
(2) Sensors use a single message to send a batch of collected measurements. The size of each message is proportional to the number of measurements collected by the policy.

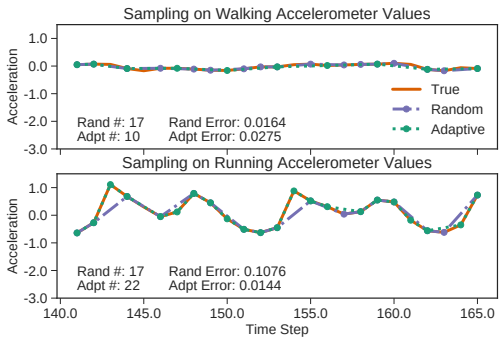


**Figure 1: An example of subsampling two accelerometer signals on the task of human activity recognition.**

# 3 PRIVACY THREATS AND ADAPTIVE SAMPLING

This section describes the considered threat model (§3.1). We then show how adaptive policies leak information within this model (§3.2) and present examples of concrete systems (§3.3).

## 3.1 Threat Model

Sensing applications often involve private information. For example, wearable medical sensors [11, 92, 99] must collect measurements without exposing a patient's diagnosis. Thus, sensors must communicate without leaking information about the captured values. Within the presented system model, we want sampling policies that minimize error and meet energy constraints, all without leaking information.

We consider a passive attacker that observes the wireless link between sensor and server using a network sniffing device. The sensor and server encrypt messages, and the attacker cannot learn anything from the ciphertext content. The adversary can instead leverage communication metadata such as message sizes and transmission times. Further, based on prior work, we assume the attacker can identify the sensor for each intercepted message [12, 84, 89, 95, 101]. We highlight that the adversary does not require physical device access. The attacker only needs to intercept the wireless communication between sensor and server. Due to the broadcast nature of wireless communication, this feature allows the adversary to operate anywhere within the range of the sensor's radio.

The adversary uses this metadata to infer sensed events. As sensors communicate at regular intervals, message times provide no useful information [101]. We instead consider attackers who infer events using message sizes (Figure 2). Using Kerckhoffs' Principle, we assume the attacker knows the set of possible events and can link multiple messages to the same (unknown) event. This assumption is realistic as the physical systems being sensed exhibit gradual changes (relative to computing speeds). Further, the attacker knows the sampling policy, as well as any employed defenses. Finally, the attacker has an offline dataset to fit a model that predicts events using batched message lengths.

We highlight two details of this threat model. First, we assume sensors do not use lossless compression as such techniques are known to leak information through message lengths (§7). Second,

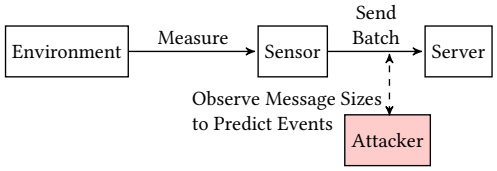


**Figure 2: Diagram of the sensor system and threat model.**

**Table 1: Average (standard deviation) message size of adaptive policies when conditioned on the underlying event.**

| Event | Linear [25] | Deviation [96] | Skip RNN [22] |
|---|---|---|---|
| Seizure | 870.12 (±241.83) | 859.00 (±286.50) | 941.76 (±233.13) |
| Walking | 564.27 (±67.50) | 489.51 (±42.14) | 784.97 (±165.11) |
| Running | 1127.46 (±65.85) | 1200.92 (±55.97) | 877.41 (±223.44) |
| Sawing | 1021.80 (±87.78) | 1080.20 (±98.85) | 1235.20 (±89.86) |

we focus on settings where each batched message corresponds to exactly one event. This setup is ideal for an attacker, as they can attribute any length variance to a single event. Our defense (§4) extends to settings where batches contain multiple events.

This threat model presents a realistic attack vector against sensors in hard-to-access places. For example, consider sensors on satellites orbiting the earth [31]. This setting deters two types of attacks. First, due to cost, an adversary cannot easily deploy devices to obtain equivalent measurements. Second, remote sensors make it difficult for an adversary to exploit side-channels that require physical device access [75, 93, 94]. Instead, the passive observation of message sizes represents a low-cost attack because the adversary can launch the attack remotely. It is thus critical to prevent adaptive sampling from leaking sensor information through its communication patterns. We emphasize that these other attack vectors against sensor networks still exist, and it remains necessary to secure both sensors and adaptive sampling from such techniques.

## 3.2 Privacy and Adaptive Sampling

Using this threat model, we observe how adaptive sampling leaks information on low-power sensors. We execute three adaptive policies (§5.1) on the task of detecting epileptic seizures from a set of activities [112]. Each policy shows a distinct message size distribution for each event, and these differences are recognizable (Table 1). For example, if a sensor is using the Deviation policy [96], an attacker who observes a 450-byte message can infer the subject is walking. This issue occurs for all events and policies; for each policy, the pairwise differences between the conditional distributions are statistically significant under a Welch's t-test ($\alpha = 0.01$). The scope of this problem indicates the issue is not with the specifics of a single policy. Instead, general-purpose adaptive sampling is *designed* to use context-dependent behavior, and this feature leads to an exploitable side-channel.

## 3.3 Example Systems

The considered system and threat models provide a general attack against adaptive sampling on low-power sensors. We discuss two systems that fit into these models.

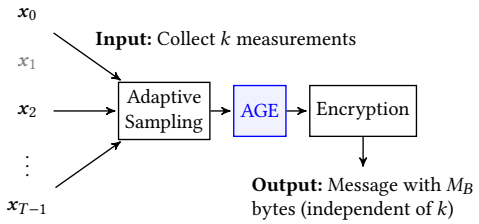


**Figure 3: AGE works in between sampling and encryption, and it requires no changes to either step.**

Nanosatellites present a cost-effective solution for remote sensing [41, 85, 87, 107]. These satellites revolve around the earth, capture measurements, and transmit values to ground base stations. This communication is periodic because it occurs when the satellite is within range of a base station. These satellites operate using battery or intermittent power, making energy management a priority. Further, the recent Orbital Edge Computing [30, 31] proposal highlights the benefits of employing adaptive behavior by only transmitting relevant measurements. This design makes the batched payload size proportional to the number of "interesting" values. An adversary can passively monitor the long-range downlink and use the payload size to possibly uncover information about the sensed values. We emphasize that it is difficult and costly to launch an attack on satellites that uses physical device contact.

ZebraNet is a sensing system for wildlife monitoring [119]. These sensors collect GPS and acceleration measurements from wild Zebras [118]. With this information, researchers can track movements and infer activities [80]. ZebraNet sensors transmit measurements every two hours to other nodes within a few kilometers. As sensors are attached to Zebras, device maintenance is costly, making energy management a key priority. Wildlife monitoring requires data privacy when tracking an endangered species. Although Zebras do not fit this category, systems such as TigerCENSE perform such monitoring [15]. When tracking endangered wildlife, it is critical to protect the location of animals from poachers. The sensors should not leak information that can localize the animal to specific areas. In this setting, adversaries will not launch an attack that requires physical device access. If the attacker could access the device, they would also have access to the target animal. An adversary can instead launch an attack by intercepting wireless communication near a centralized base station.

# 4 ADAPTIVE GROUP ENCODING

Adaptive Group Encoding (AGE) is a novel framework to protect adaptive sampling policies from leaking information through the size of batched messages. AGE creates fixed-length messages for all collected batches, breaking the relationship between message size and collection rate. The framework operates as a drop-in step between sampling policy and encryption module, allowing AGE to work with any adaptive sampler, sensing task, and encryption algorithm (Figure 3). Achieving this generality prevents AGE from making strong assumptions about the sampling policy and the measurement values. AGE only requires that sensors capture numerical data. Within this setting, AGE must use lossy encoding; there is no

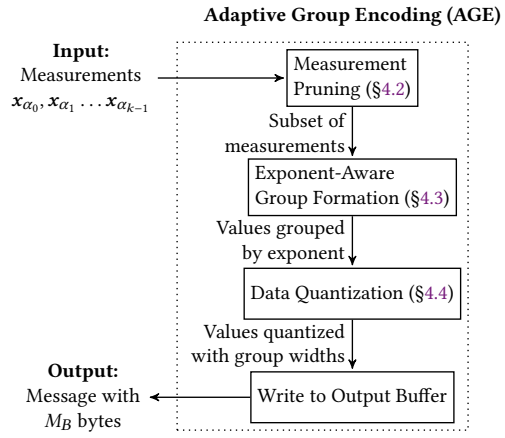


**Figure 4: Overview of the data flow through AGE.**

lossless algorithm that can guarantee fixed-length messages [69]. Fortunately, this design aligns with existing sampling techniques; sampling is already lossy because it drops entire measurements. AGE aims to minimize the additional error required to prevent information leakage.

Central to AGE is fixed-point quantization, a cheap technique that can encode measurements into same-sized messages by properly setting the bit width for each value. Quantization alone, however, faces three issues that lead to high error. First, the sampler may capture more values than available bits, forcing this encoder to drop all elements to meet the target size. Second, quantization uses a static number of non-fractional bits, causing unnecessary errors by not adapting to data ranges. Finally, quantization must round the selected width down to the nearest integer, resulting in excessive message padding.

AGE addresses these shortcomings by applying three transformations (Figure 4). First, AGE introduces a pruning step that removes measurements from batches with too many values (§4.2). Second, AGE adapts to the observed values by supporting dynamic ranges (§4.3). The framework applies run-length encoding (RLE) to compress value exponents. This RLE step groups adjacent values with the same exponent. AGE assigns bit widths for each exponent-aware group and quantizes the enclosed values accordingly (§4.4). Working with these smaller sets allows AGE to better utilize the available message bytes.

AGE protects adaptive sampling executing on MCUs under energy constraints. We create AGE to have a small energy cost, and AGE offsets any computational overhead by reducing the amount of wireless communication (§4.5).

## 4.1 Notation

We introduce the relevant notation before describing the AGE framework. As MCUs work in fixed-point arithmetic, we consider measurements $\boldsymbol{x}_t \in \mathbb{R}^d$ encoded as fixed-point values. Dach value has $w_0$ bits, $n_0$ of which are non-fractional places. The term $n_0$ logically corresponds to the value's exponent, as the binary point is in the $(w_0 - n_0)^{th}$ place. We consider a general adaptive sampler operating under an energy budget of $B$ joules; the policy meets this constraint using an average collection rate $\rho_B$. AGE knows $B$ and, by extension, $\rho_B$. The sensor has a sampling period of $\Delta t$ seconds and sends measurement batches every $\Delta T$ seconds (§2.1). The sensor sends at most $T = \Delta T / \Delta t$ measurements in every batch. We denote the target message size by $M_B$; this term is the number of bytes required to encode $\lfloor \rho_B \cdot T \cdot d \rfloor$ values. The target size $M_B$ is proportional to the budget $B$ through the collection rate $\rho_B$. For brevity, $M_B$ does not include the fixed amount of metadata required for communication. In practice, AGE handles this overhead by reducing the space available for data. Finally, $k$ is the number of captured measurements in a single batch, and $\alpha_t$ is the index of the $t^{th}$ collected element. Table 2 summarizes this notation.

**Table 2: A selection of relevant notation.**

| Term | Description |
|---|---|
| $T$ | The maximum number of elements per batch. |
| $\boldsymbol{x}_t$ | The measurement at step $t$. |
| $d$ | The number of features in each measurement. |
| $B$ | The energy budget in joules. |
| $k$ | The number of elements captured by the sampling policy. |
| $M_B$ | The target message size for the budget $B$. |
| $\rho_B$ | The average collection rate to meet the budget $B$. |
| $\alpha_t$ | The original index of the $t^{th}$ collected sample. |
| $w_0$ | The original number of bits per feature. |
| $n_0$ | The original number of non-fractional bits per feature. |

## 4.2 Measurement Pruning

AGE ensures fixed-length messages for any periodic adaptive sampling policy. Hence, AGE cannot make strong assumptions about the policy's collection patterns. Even under tight energy constraints, the base policy may capture all $k = T$ elements, yet AGE must still create a message with $M_B$ bytes.

Fixed-point quantization alone leads to poor results for scenarios requiring a high compression ratio. For example, consider when $M_B = 35$, $k = 50$ and $d = 6$. In this setting, dedicating one bit per value yields a message with 38 bytes, eclipsing the target $M_B = 35$. For quantization to meet the target, it must drop all values.

AGE achieves better error during extreme over-sampling by taking a more moderate approach. AGE instead removes just enough measurements to ensure that all values get at least $w_{min}$ bits. AGE performs this pruning by first computing the distance scores between consecutive measurements.

$$\texttt{Dist}(\boldsymbol{x}_{\alpha_t}) = \|\boldsymbol{x}_{\alpha_t} - \boldsymbol{x}_{\alpha_{t+1}}\|_1 + \frac{1}{8} \cdot |\alpha_t - \alpha_{t+1}| \tag{1}$$

AGE removes the $\ell$ (see below) measurements $\boldsymbol{x}_{\alpha_t}$ which have the smallest Dist. This metric estimates the error caused by dropping $\boldsymbol{x}_{\alpha_t}$. By including the time difference, AGE avoids creating long stretches with no collected values. Long gaps often lead to high error because the system cannot detect signal changes in the interim [63]. The factor of $\frac{1}{8}$ enables scaling using cheap bit shifting.

AGE sets $\ell$ as the largest positive integer such that $\lceil \frac{1}{8} \cdot w_{min} \cdot (k-\ell) \cdot d \rceil \leq M_B$. If no such $\ell > 0$ exists, AGE skips the pruning step. This setting ensures that all remaining values get represented with at least $w_{min}$ bits. In our experiments, we set $w_{min} = 5$; a smaller minimum leads to higher error during quantization (§4.4). We note

that incrementally updating the Dist scores yields an algorithm with lower error, but we find the overhead is not worth the benefits.

### 4.3 Exponent-Aware Group Formation

When the adaptive policy over-samples, AGE must compress values into a message with $M_B$ bytes. AGE performs this compression using quantization (§4.4). By itself, fixed-point quantization uses a static number of non-fractional places to limit the error for large values; this strategy causes unnecessary errors by not adapting to data ranges. For example, let $w_0 = 7$ and $n_0 = 5$, and consider quantizing to 3 bits. A static quantizer encodes 1.5 as 0 because the 0 and 4 are the two closest representable values in the 3-bit format. To avoid such errors, it is critical to enable dynamic ranges.

The challenge is that the optimal number of non-fractional bits is value-dependent. In the example above, 1.5 requires $n = 2$ non-fractional places in the 3-bit format while 0.25 needs $n = 1$. Providing each value with its own exponent takes considerable space, where these exponents store the number of non-fractional bits. AGE manages this tension by noting that adjacent sensor values often fall in similar ranges [90]. Thus, AGE can compress exponents with run-length encoding (RLE). This process creates measurement groups consisting of adjacent values with the same exponent.

RLE alone provides no guarantees on its compression ratio. In the worst case, the encoded exponents can exceed $M_B$ bytes. Thus, AGE must cap the maximum space for the exponents using a merging step. This process scores adjacent groups $g_1$ and $g_2$ with non-fractional places $n_1$ and $n_2$ as defined below.

$$\texttt{Score}(g_1, g_2) = \texttt{Count}(g_1) + \texttt{Count}(g_2) + 2 \cdot |n_1 - n_2| \quad (2)$$

AGE greedily merges groups with the lowest initial scores to produce at most $G$ sets (see below). Upon merging, the new group uses $\max(n_1, n_2)$ non-fractional places to minimize the error of large values. The factor of two balances the impact of exponent values and group counts, and we implement this scaling using cheap bit shifts. AGE skips this merging step when the initial number of groups is less than $G$. We note that an algorithm that updates scores after each merge yields a better approximation. The benefits of this approach, however, are not worth the overhead on an MCU.

The maximum number of groups $G$ controls the tradeoff between exponent fidelity and space overhead. AGE sets this parameter by first finding the number of bytes $m$ required if each value were encoded using the full $w_0$ bits. AGE dedicates the remaining $M_B - m$ bytes to exponents. That is, AGE sets $G$ to the greatest number of groups whose metadata fits within $M_B - m$ bytes. We enforce that $G = \max(G, G_0)$ to limit the exponent approximation error when $k > \rho_B T$. By expanding the number of groups when possible, AGE reduces space wasted on padding when the policy under-samples. We use $G_0 = 6$ in our experiments, though we find that AGE's performance is not sensitive across $G_0 = 4, 6, 8$.

### 4.4 Data Quantization

AGE leverages the exponent-aware groups to quantize values and control the message length. In particular, AGE sets the bit width of the values in each group such that the encoded result is at most $M_B$ bytes. This group-based strategy addresses a limitation of standard fixed-point quantization. The standard encoder will use the width $w = \left\lfloor \frac{8 \cdot M_B}{k \cdot d} \right\rfloor$ to ensure a result of at most $M_B$ bytes. This system must round the width down, leading to wasted space. For example, let $M_B = 220, k = 50$ and $d = 6$. These settings yield a width of $w = 5$, corresponding to 188 data bytes. This result means that over 14% of the message gets wasted padding up to $M_B = 220$ bytes.

AGE uses its exponent-aware groups (§4.3) to improve this utilization. The framework gives each group a bit width to use when quantizing its enclosed values. AGE sets these widths using a round-robin process to optimize the number of bytes used under the message size $M_B$. This strategy works better than a uniform assignment because the groups enable adjustments in smaller multiples. For instance, using 5 groups of 10 measurements in the previous example, AGE will assign 5 bits to one group and 6 bits to the remaining four. This assignment creates a message with 218 data bytes; only 1% of the message gets wasted on padding. The per-group assignments allow AGE to functionally mimic fractional bit widths.

AGE's per-group quantization requires encoding the widths into the message. This overhead is manageable because the widths are small and only need a few bits each—e.g., four bits when $w_0 = 16$. Further, the maximum number of groups $G$ places a cap on the total number of widths. We note that AGE accounts for the space required to store exponents, group sizes, and bit widths when selecting the quantization parameters.

In summary, AGE quantizes values in the $i^{th}$ group as fixed-point integers with $w_i$ bits, $n_i$ of which are non-fractional places (§4.3). AGE packs the quantized values, as well as the group metadata, into an output buffer. This design allows AGE to optimize its use of the available $M_B$ bytes and preserve the benefits of dynamic ranges.

### 4.5 Discussion

AGE uses a multi-step process to minimize its encoding error. This algorithm is more expensive than a standard procedure which directly packs values into an output buffer (§5.8). We address this overhead by saving energy on wireless communication. AGE produces messages with $M_B$ bytes, and the framework can counteract any encoding overhead by reducing this target size. Intuitively, AGE spends a bit more energy on computation and saves significant energy on communication. In practice, we reduce the target length by about 30 bytes and include an additional 20-byte reduction for every 500-byte multiple in $M_B$. This conservative estimate allows AGE to display negligible energy overhead in deployment (§5.7).

We emphasize the generality of the presented system. In particular, AGE works for both block and stream ciphers. For block ciphers, AGE uses the target size $M_B$ rounded to the nearest block. In the case of a stream cipher, AGE uses the size $M_B$ as given. The only knowledge AGE requires of the encryption algorithm is the size of associated metadata (e.g. nonces). These quantities allow AGE to determine the number of bytes available for measurements.

As a final point, AGE ensures that the sent message has $M_B$ bytes absent any external faults. This guarantee is similar to those provided by systems with hard real-time constraints [103]. For example, if the network drops a packet, an attacker may observe a message of size $\tilde{M} < M_B$. AGE assumes that such faults occur independently of the sensed events. Thus, an attacker cannot use intermittent failures as a source of information leakage.

# 5 EVALUATION

AGE protects low-power adaptive policies from leaking information through message sizes. We evaluate AGE by measuring its error, information leakage, and energy consumption. This evaluation demonstrates the following.

(1) Under strict energy budgets, adaptive policies with AGE achieve lower error than non-adaptive sampling and defenses using message padding (§5.2).
(2) General adaptive sampling policies leak information through messages sizes. By standardizing message lengths, AGE protects multiple adaptive policies from this leakage both in theory (§5.3) and in practice (§5.4).
(3) AGE's generality allows it to protect a variety of adaptive policies, including strategies based on trainable models such as neural networks (§5.5).
(4) AGE uses multiple transformations to minimize its error. With this design, AGE obtains a lower error than variants using quantization and pruning alone (§5.6).
(5) On a resource-constrained MCU, AGE retains the low error of adaptive sampling and eliminates all information leakage through message sizes (§5.7). These benefits come with negligible energy overhead (§5.8).

## 5.1 Setup

We evaluate AGE both in simulation and on a low-power MCU. We use nine sensor datasets (Table 3) that include both floating-point and integer measurements. Each sequence constitutes a batch, and the batches range from 98 to 3, 138 bytes. We highlight two datasets for their relation to the examples in §3.3. The Activity [8] task uses accelerometer values to classify human activities; these measurements are similar to those of ZebraNet [119], albeit on humans instead of wildlife. The Tiselac [55] dataset uses satellite image features to infer land cover types; this task relates to nanosatellites.

We execute each policy under energy constraints. Upon budget violation, the policy uses random values for the remaining sequences. The server receives a subsequence and infers missing elements using linear interpolation. We measure the mean absolute error (MAE) of this reconstruction. We use eight budgets per dataset corresponding to the energy of a Uniform sampler capturing 30%, 40%, …, 100% of elements. The simulator tracks energy using traces from a TI MSP430 FR5994 [3]. We conservatively multiply AGE's energy cost by 4×. The simulator results use a ChaCha20 stream cipher (IETF RFC 7539). For brevity, we omit the block cipher results. AGE provides equivalent protection with block ciphers, and we find AGE has a better relative error in this setting due to the extra bytes available from block padding the target size.

The hardware environment consists of a TI MSP430 FR5994 MCU [3] with an HM-10 BLE module. The MCU performs sensing by reading measurements from FRAM, and the system processes one sequence every six seconds. We measure the energy consumption of each policy using the EnergyTrace™ tool [2]. We use an AES-128 block cipher [35] because the MCU has an AES accelerator.

For each adaptive policy, we compare AGE to two alternative approaches. The first is the standard adaptive policy with no post-processing step. The second baseline uses a message padding algorithm analogous to BuFLO [36]. We use the minimum padding amount by assuming the defense knows the largest batch within the evaluation data.

**Table 3: Properties of the evaluation datasets.**

| Dataset | # Seq | Seq Len | # Feat | Labels | Bits (Frac) | Range |
|---|---|---|---|---|---|---|
| Activity [8] | 11,119 | 50 | 6 | 12 | 16 (13) | 10.6 |
| Characters [116] | 1,436 | 100 | 3 | 20 | 16 (13) | 7.8 |
| EOG [37] | 362 | 1,250 | 1 | 12 | 20 (8) | 2640.4 |
| Epilepsy [112] | 138 | 206 | 3 | 4 | 16 (13) | 7.2 |
| MNIST [64] | 10,000 | 784 | 1 | 10 | 9 (0) | 255 |
| Password [1] | 308 | 1,092 | 1 | 5 | 16 (11) | 18.8 |
| Pavement [100] | 8,864 | 120 | 1 | 3 | 16 (10) | 68.4 |
| Strawberry [53] | 370 | 235 | 1 | 2 | 16 (13) | 5.9 |
| Tiselac [55] | 17,973 | 23 | 10 | 9 | 16 (0) | 3379 |

We experiment with one non-adaptive policy (Uniform) and three adaptive sampling algorithms (Linear, Deviation, and Skip RNN). We also try a Random sampling baseline, but we omit these results because Uniform policy displays better error. We evaluate AGE on all adaptive policies.

**Uniform** The Uniform sampling algorithm collects $k$ elements from each sequence where $k$ is the maximum amount adhering to the budget $B$. This policy collects indices $t = r\lceil T/k \rceil$ for $i \in \{0, 1, \ldots, k-1\}$. When $k$ does not divide $T$, we include additional random indices to ensure the policy collects $k$ elements.

**Linear** The Linear adaptive policy [25] alters its collection rate based on the differences in consecutive measurements. When the absolute difference exceeds a threshold, the policy collects the next element. Otherwise, it increases its collection period by one. We use an offline training step to set a threshold for each budget.

**Deviation** The Deviation adaptive policy [96] uses a weighted moving average to track the measurement variance. When the average variance exceeds the threshold, the policy doubles its collection rate. Otherwise, the policy halves its rate. We set the per-budget thresholds using an offline process.

**Skip RNN** The adaptive Skip RNN algorithm [22] uses offline data to train a recurrent neural network (RNN) which learns to sample. As Skip RNNs consume more energy than the other policies, we do not use this sampler under energy budgets. We instead use Skip RNNs to display the generality of AGE and its applicability to near-future adaptive sampling. This application shows how AGE can protect adaptive neural networks that support subsampling [57, 81].

## 5.2 Reconstruction Error

Sampling policies aim to minimize error and meet energy constraints. Thus, AGE cannot have a high error cost. We evaluate this cost by measuring the adaptive policies' error with AGE (Table 4). These results have three main takeaways.

First, AGE retains adaptive policies' dominance over non-adaptive sampling. On all tasks, the best adaptive policy with AGE has a lower error than Uniform sampling, and AGE shows roughly 13.4% (Linear) and 11.3% (Deviation) lower error overall. Further, AGE has better error than Uniform sampling on 73.6% (Linear) and 70.8% (Deviation) of constraints, and AGE makes adaptive sampling worse than the Uniform policy on only one additional budget. On some tasks, policies with AGE dominate Uniform sampling (Figure 5).

Second, unlike message padding, AGE meets the energy constraints of low-power devices. The Padded defense has a high error

**Table 4: Arithmetic mean reconstruction error (MAE) across all budgets. The asterisk denotes the lowest error overall, and the boldface marks the lowest error for policies without information leakage. The final row is the median percent error higher than Uniform sampling (lower is better).**

| Dataset | Unif. | Linear | | | Deviation | | |
|---|---|---|---|---|---|---|---|
| | | *Std* | *Padded* | *AGE* | *Std* | *Padded* | *AGE* |
| Activity | 0.0146 | 0.0090* | 0.0565 | **0.0095** | 0.0099 | 0.0622 | 0.0104 |
| Characters | 0.0046 | 0.0046 | 0.0404 | 0.0046 | 0.0045* | 0.0443 | **0.0046** |
| EOG | 0.1343 | 0.1251* | 31.7824 | **0.1259** | 0.1301 | 37.7492 | 0.1321 |
| Epilepsy | 0.1090 | 0.0992* | 0.2076 | **0.0997** | 0.0998 | 0.2295 | 0.1005 |
| MNIST | 5.0770 | 4.9231 | 6.5239 | 4.9397 | 4.6694* | 7.1768 | **4.6958** |
| Password | 0.0073 | 0.0024* | 0.1002 | **0.0024** | 0.0026 | 0.1063 | 0.0026 |
| Pavement | 0.7594 | 0.6477 | 0.7233 | 0.6886 | 0.6301* | 0.7228 | **0.6786** |
| Strawberry | 0.0059 | 0.0049 | 0.0320 | 0.0050 | 0.0048* | 0.0415 | **0.0049** |
| Tiselac | 2.7539 | 2.6547* | 9.5832 | **2.6770** | 2.7762 | 9.5012 | 2.7934 |
| Overall (%) | 0.00 | -15.84* | 135.43 | **-13.41** | -15.18 | 137.74 | -11.34 |

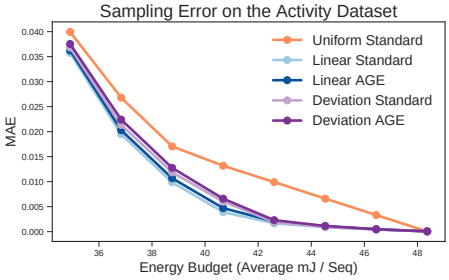


**Figure 5: MAE for each budget on the Activity dataset.**

on all tasks due to budget violations caused by extra communication. In contrast, AGE never exceeds the budget because it decreases the amount of wireless communication to limit its energy overhead (§4.5). This design allows AGE to achieve equal or better error than Padded policies on 94.4% of budgets. Amongst the policies which leak no information (§5.3), AGE provides the best overall error.

Third, for adaptive policies, AGE does incur a cost in error. In particular, adaptive sampling with AGE displays a median of about 0.92% higher error than their standard counterparts. We expect this additional error based on the lossy approach employed by AGE; however, we believe this very small error is a small price for eliminating information leakage.

AGE requires the highest compression ratio when the underlying policy captures the most elements. This phenomenon generally occurs on sequences with high variation. To ensure that AGE's error is not prohibitive in these situations, we supplement the MAE values with a weighted error metric based on sequence deviation. Specifically, we weigh the MAE from each sequence by the standard deviation of its measurement values. Table 5 shows the weighted error values averaged across all budgets. AGE continues to consistently outperform Uniform sampling. AGE achieves a lower weighted error on every dataset, leading to a 15% lower median error than the Uniform policy. This aggregate result eclipses the unweighted value (Table 4) due to the better performance of adaptive sampling on high-deviation sequences. Furthermore, on all datasets but one, AGE is the best-performing policy that protects against information leakage. On the Pavement task, the Padded variant achieves a lower

**Table 5: Arithmetic mean weighted reconstruction error across all budgets. The asterisk marks the lowest overall error, and the boldface shows the lowest error amongst policies with no leakage.**

| Dataset | Unif. | Linear | | | Deviation | | |
|---|---|---|---|---|---|---|---|
| | | *Std* | *Padded* | *AGE* | *Std* | *Padded* | *AGE* |
| Activity | 0.0274 | 0.0134* | 0.0652 | **0.0144** | 0.0147 | 0.0756 | 0.0159 |
| Characters | 0.0049 | 0.0047 | 0.0343 | 0.0048 | 0.0046* | 0.0376 | **0.0047** |
| EOG | 0.1382 | 0.1205* | 40.4317 | **0.1274** | 0.1243 | 46.3721 | 0.1392 |
| Epilepsy | 0.1394 | 0.1067 | 0.1993 | 0.1073 | 0.1027* | 0.2299 | **0.1039** |
| MNIST | 5.1211 | 5.0021 | 6.5698 | 5.0206 | 4.7047* | 7.1624 | **4.7344** |
| Password | 0.0073 | 0.0024* | 0.1002 | **0.0024** | 0.0026 | 0.1063 | 0.0026 |
| Pavement | 0.8919 | 0.6835 | 0.7185 | 0.7451 | 0.6261* | **0.6693** | 0.7019 |
| Strawberry | 0.0059 | 0.0049 | 0.0320 | 0.0050 | 0.0048* | 0.0415 | **0.0049** |
| Tiselac | 4.6389 | 4.5302* | 12.1781 | **4.5784** | 4.8623 | 12.2598 | 4.9076 |
| Overall (%) | 0.00 | -16.95 | 162.52 | -15.25 | -18.64* | 175.91 | **-16.95** |

**Table 6: Median / maximum empirical normalized mutual information between message size and event label. Padded and AGE have the same median and maximum values.**

| Dataset | Linear | | | Deviation | | |
|---|---|---|---|---|---|---|
| | *Std* | *Padded* | *AGE* | *Std* | *Padded* | *AGE* |
| Activity | 0.34 / 0.35 | 0.00 | 0.00 | 0.33 / 0.40 | 0.00 | 0.00 |
| Characters | 0.24 / 0.25 | 0.00 | 0.00 | 0.24 / 0.26 | 0.00 | 0.00 |
| EOG | 0.20 / 0.30 | 0.00 | 0.00 | 0.19 / 0.31 | 0.00 | 0.00 |
| Epilepsy | 0.41 / 0.44 | 0.00 | 0.00 | 0.39 / 0.47 | 0.00 | 0.00 |
| MNIST | 0.13 / 0.13 | 0.00 | 0.00 | 0.14 / 0.27 | 0.00 | 0.00 |
| Password | 0.07 / 0.10 | 0.00 | 0.00 | 0.09 / 0.12 | 0.00 | 0.00 |
| Pavement | 0.13 / 0.13 | 0.00 | 0.00 | 0.13 / 0.15 | 0.00 | 0.00 |
| Strawberry | 0.05 / 0.09 | 0.00 | 0.00 | 0.06 / 0.09 | 0.00 | 0.00 |
| Tiselac | 0.07 / 0.12 | 0.00 | 0.00 | 0.11 / 0.20 | 0.00 | 0.00 |

error than AGE. This result occurs because the sequences after the budget violation exhibit low deviation. Thus, the sequences that require random guessing get less weight, leading to a lower aggregate error value. This result is a product of the arbitrary dataset ordering and is not a fundamental benefit of the Padding strategy. The far superior performance of AGE on the other eight datasets shows the benefits of the proposed approach. In general, the low weighted error shows how AGE outperforms baseline approaches when focusing on scenarios that require the highest compression from the lossy encoding routine.

## 5.3 Information Leakage: Theoretical

AGE protects adaptive sampling from leaking information through message lengths. We demonstrate this ability from an information-theoretic perspective (§5.4 explores practical implications). For each budget, we estimate the normalized mutual information (NMI) between the event $L$ and the message size $M$ (see below) [62]. This metric represents the reduction in uncertainty about the event after observing the message size [26]. The terms $\hat{I}(L,M)$, $\hat{H}(L)$, and $\hat{H}(M)$ are the maximum likelihood estimators for the mutual information and (Shannon) entropy.

$$\text{NMI}(L,M) = \frac{2 \cdot \hat{I}(L,M)}{\hat{H}(L) + \hat{H}(M)} \tag{3}$$

A policy with no leakage should have zero mutual information, implying that the message size is independent of the event.

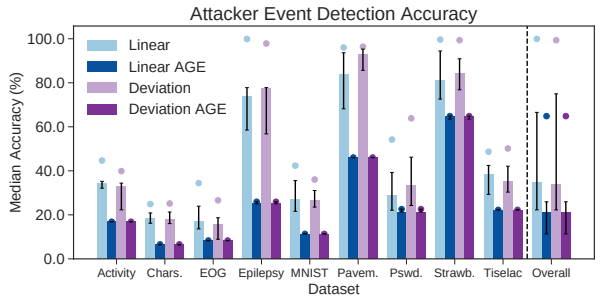


**Figure 6: Median attacker accuracy across all energy budgets. The error bars denote the first and third quartiles, and the points show the maximum accuracy.**

| Tr \ Pr | Seizure | Other |
|---|---|---|
| *Seizure* | 500 | 0 |
| *Other* | 0 | 1500 |

**(a) Linear**

| Tr \ Pr | Seizure | Other |
|---|---|---|
| *Seizure* | 0 | 500 |
| *Other* | 0 | 1500 |

**(b) Linear with AGE**

**Figure 7: Confusion matrices for the attack model under a single budget. The Linear policy yields 100% accuracy on detecting seizures. AGE forces all predictions into one event.**

The Uniform policy has zero NMI because the policy captures the same number of elements for all events. In contrast, *both* standard adaptive policies display nonzero NMI across all tasks (Table 6), indicating that the size of messages from adaptive policies leaks information about event labels. We estimate the significance of this leakage using an approximate permutation test [48]. This test randomly shuffles the message lengths and recomputes the NMI. With this design, the null hypothesis is that random variation in the lengths leads to the nonzero NMI instead of any true dependence between sizes and events. We use 15,000 permutations for each test, creating a worst-case 95% confidence interval of $\hat{p} \pm 1.96 \cdot \frac{1}{2\sqrt{15000}}$ where $\hat{p}$ is the estimated $p$-value [82]. With this methodology, we find that the adaptive policies have an entire 95% $p$-value confidence interval less than 0.01 on 83.3% (Linear) and 81.9% (Deviation) of evaluated budgets. This result shows that, with a high likelihood, the observed NMI occurs due to a link between message lengths and event labels. Thus, the observed NMI corresponds to a meaningful amount of leakage. We emphasize that this leakage occurs for both policies and across all tasks, showing the scope of the privacy issue.

AGE successfully eliminates the information leakage of adaptive sampling. As a result of same-sized messages, adaptive policies with AGE show zero NMI between message length and event label. Thus, AGE ensures that the message size is independent of the event. This result holds despite the base adaptive policy using data-dependent collection rates. Further, this protection applies to both adaptive policies across all tasks, supporting the generality of this defense. We note that Padding provides equivalent security, but it does so at a much higher energy cost (§5.7).

**Table 7: Average MAE, maximum NMI, and maximum attack accuracy when sampling using Skip RNNs.**

| Dataset | MAE | | NMI | | Attack (%) | |
|---|---|---|---|---|---|---|
| | *Skip RNN* | *AGE* | *Skip RNN* | *AGE* | *Skip RNN* | *AGE* |
| Activity | 0.0081 | 0.0087 | 0.40 | 0.00 | 59.26 | 16.87 |
| Characters | 0.0043 | 0.0044 | 0.34 | 0.00 | 26.60 | 6.59 |
| EOG | 0.1434 | 0.1439 | 0.39 | 0.00 | 35.15 | 8.33 |
| Epilepsy | 0.0892 | 0.0896 | 0.31 | 0.00 | 92.80 | 25.95 |
| MNIST | 0.9192 | 1.0207 | 0.13 | 0.00 | 44.70 | 11.35 |
| Password | 0.0073 | 0.0073 | 0.08 | 0.00 | 43.82 | 22.60 |
| Pavement | 0.5845 | 0.6091 | 0.20 | 0.00 | 96.45 | 46.05 |
| Strawberry | 0.0040 | 0.0040 | 0.09 | 0.00 | 97.70 | 64.85 |
| Tiselac | 2.0832 | 2.2810 | 0.16 | 0.00 | 54.73 | 22.26 |

## 5.4 Information Leakage: Practical

The NMI results provide a strong indication that AGE protects standard adaptive policies from leaking information through message lengths. We supplement this theoretical analysis by presenting a practical attack. We consider an adversary who uses message lengths of ten random sequences of the same event (§3.1). The input features include the average, median, standard deviation, and IQR of the message sizes. We use an ensemble of 50 decision trees fit with AdaBoost [40, 91] and measure the test accuracy using stratified five-fold cross-validation. The training and testing sets contain 8,000 and 2,000 samples, respectively. Policies with no leakage should have a test accuracy equivalent to the most frequent event. As this attack uses one methodology, the presented approach is a lower bound for what an adversary may uncover.

With this attack, the adversary can infer events from standard adaptive policies (Figure 6). The adaptive policies have an overall median attack accuracy that is 1.58× higher than the most frequent event. In the worst case, the policies allow the attacker to infer over 94% of the event labels on the Epilepsy, Pavement, and Strawberry datasets. This result occurs for both the Linear and the Deviation policies, confirming the privacy problem associated with general-purpose adaptive sampling. We emphasize that this worse-case leakage occurs for sensitive events. For example, the attack model against the Linear policy can obtain 100% precision and recall on classifying the occurrence of an epileptic seizure (Figure 7a).

AGE fixes this issue for both adaptive policies. Due to fixed-length messages, both policies with AGE display an attack accuracy equivalent to the most frequent label, the best that is practically achievable. We highlight this behavior on the previous scenario for adaptive sampling (Figure 7). AGE's same-sized messages force all predictions into a single event, thereby fully protecting adaptive sampling even in worst-case situations (Figure 7b). Overall, AGE eliminates the information leakage problem from adaptive policies, meets energy requirements, and retains a low error (Table 4).

## 5.5 Evaluation on Skip RNNs

AGE works as a post-processing step to adaptive sampling, making AGE compatible with any policy. We display this generality by applying AGE to Skip RNNs [22]. We evaluate these models when collecting 30%, 40%, …, 100% of elements.

As shown in Table 7, AGE incurs roughly 1.60% greater error than standard Skip RNNs. The Skip RNN policy, however, leaks information on each task. With a permutation test, the observed nonzero

**Table 8: Median percent error greater than AGE across all budgets and tasks. Higher values indicate higher error.**

| Variant | Linear | Deviation |
|---|---|---|
| Single | 2.664% | 2.870% |
| Unshifted | 2.068% | 2.548 % |
| Pruned | 58.882% | 57.278% |
| AGE | **0.000%** | **0.000%** |

**Table 9: Average energy per sequence (mJ) over 75 sequences when on a TI MSP430 MCU under three energy budgets.**

| Policy | Activity | | | Tiselac | | |
|---|---|---|---|---|---|---|
| | 2.837J | 3.285J | 3.634J | 2.606J | 2.929J | 3.288J |
| Uniform | 37.83 | 43.81 | 48.46 | 34.75 | 39.05 | 43.85 |
| Linear | 37.22 | 42.86 | 48.18 | 34.84 | 38.93 | 43.75 |
| Padded | 45.35 | 48.21 | 48.22 | 37.38 | 43.87 | 43.67 |
| AGE | 36.37 | 42.62 | 47.09 | 34.46 | 37.59 | 43.63 |
| Deviation | 37.30 | 42.68 | 48.17 | 35.01 | 39.18 | 43.75 |
| Padded | 48.36 | 48.11 | 48.60 | 37.28 | 43.50 | 43.70 |
| AGE | 36.31 | 42.01 | 47.00 | 34.52 | 37.90 | 43.61 |

NMI values are significant on over 72.2% of rates. Furthermore, this leakage translates into a practical attack with a worst-case accuracy of over 95%. This result further supports that general-purpose adaptive sampling algorithms leak information on multiple tasks. AGE successfully closes this side-channel. Skip RNNs with AGE show zero NMI between message size and event, and the attack accuracy gets reduced to the most frequent label. The successful application of AGE to a diverse set of policies shows the framework's generality and small performance overhead.

## 5.6 Variants of Adaptive Group Encoding

AGE performs encoding by compressing exponents and quantizing values. We evaluate these transformations using three variants. The Single variant uses fixed-point quantization with a single bit-width. The Unshifted variant uses six even-sized groups but fixes the exponent for all values. The Pruned variant removes measurements to control the message size (§4.2). All three variants create fixed-length messages, so we focus this comparison on sampling error. Across all tasks, AGE displays lower error than these baselines (Table 8). Further, AGE shows an equal or better error than all variants on over 98% of budgets. These results show how all AGE's features are necessary to achieve low error.

We highlight an important aspect of these results: quantization alone does not retain adaptive sampling's dominance over Uniform sampling. The Single and Unshifted policies incur high errors on the Tiselac task, showing MAEs of 6.245 (Single) and 8.998 (Unshifted) with the Linear policy. This error exceeds that of Uniform sampling by over 2.2× (Table 4). In contrast, AGE outperforms Uniform sampling on this task. Unlike encoding using fixed-point quantization alone, AGE allows adaptive policies to dominate Uniform sampling.

## 5.7 Performance on a Microcontroller

In simulation, AGE incurs minimal error, ensures fixed-length messages, and satisfies energy constraints. We validate these results on a TI MSP430 FR5994 [3] with an HM-10 BLE radio. We run each policy over 75 sequences on the Activity and Tiselac tasks. We set three budgets using Uniform policy's energy when collecting 40%, 70%, and 100% of elements. We enforce budget violations when the energy per sequence is significantly higher than Uniform sampling. We assess significance using a one-sided Welch's t-test ($\alpha = 0.05$).

**Table 10: MAE over 75 sequences on a TI MSP430 MCU under three energy budgets.**

| Policy | Activity | | | Tiselac | | |
|---|---|---|---|---|---|---|
| | 2.837J | 3.285J | 3.634J | 2.606J | 2.929J | 3.288J |
| Uniform | 0.0280 | 0.0103 | 0.0000 | 3.4848 | 1.5383 | 0.0000 |
| Linear | 0.0203 | 0.0016 | 0.0000 | 4.0615 | 0.9974 | 0.0000 |
| Padded | 0.0720 | 0.0306 | 0.0000 | 15.0244 | 19.2171 | 0.0000 |
| AGE | 0.0223 | 0.0023 | 0.0001 | 4.0615 | 1.0005 | 0.0000 |
| Deviation | 0.0231 | 0.0019 | 0.0001 | 4.0046 | 1.1654 | 0.0000 |
| Padded | 0.0916 | 0.0309 | 0.0001 | 14.9644 | 17.3346 | 0.0000 |
| AGE | 0.0254 | 0.0027 | 0.0002 | 4.0052 | 1.1709 | 0.0000 |

Across all budgets, AGE shows a lower average energy per sequence than both Uniform sampling and standard adaptive policies (Table 9). These results confirm the negligible energy overhead associated with AGE. This performance contrasts with the Padded policies; AGE requires a median of 9.70% (Linear) and 10.04% (Deviation) less energy per sequence than Padding. This minimal overhead comes at little cost in error (Table 10). On the MCU, AGE shows the same error under these three constraints as we observe in simulation. AGE shows a far higher error than Uniform sampling on only one budget (Tiselac at 2.606J), and this result comes from higher error by the base adaptive policies. The Padded variants show high errors due to budget violations.

Standard adaptive sampling continues to show batch size variance. On the Activity task, adaptive sampling has a median of 0.088 (Linear) and 0.101 (Deviation) NMI between message size and event label. AGE instead shows zero NMI by always sending fixed-length messages. These results show how AGE successfully protects adaptive policies and meets the constraints of low-power sensors.

## 5.8 Overhead Analysis

AGE's multi-step encoding leads to computational overhead. We analyze how this computation translates to energy on a TI MSP430 FR5994 MCU [3]. When encoding a full sequence from the Activity dataset, AGE consumes roughly 0.154mJ. This figure compares to the 0.016mJ needed for a standard process that writes values directly into an output buffer. To offset this cost, AGE reduces the amount of communication by roughly 30 bytes per batch (§4.5). For an HM-10 BLE radio, this reduction saves about 0.9mJ, far eclipsing the energy required for encoding. AGE further protects against any worse-case scaling by reducing the communication by an additional 20 bytes every ∼250 measurement values. With this design, AGE trades some computational overhead for greatly reduced communication, leading to negligible total energy overhead in practice (§5.7)

# 6 RELATED WORK

**Security in IoT and Cyber-Physical Systems** Attacks such as the Mirai botnet [10] highlight the importance of low-power device security [102, 109]. Previous systems examine the security of smart homes [4, 77] and smart cities [44, 65]. Additional work fingerprints

wireless devices [16, 18, 39, 42, 84, 89, 95]. AGE builds on this work by protecting against attackers who can fingerprint sensors.

Multiple attacks extract information through device side-channels such as electromagnetic waves [61, 75, 93, 94] and cache timing [105]. AGE also addresses a side-channel of low-power devices, but our work focuses on message lengths. The MoLe attack uses a smartwatch accelerometer to infer typed words [113]. This attack highlights the need for systems such as AGE which protect sensor measurements. RCAD protects sensor networks from leaking measurement times [56]. AGE also addresses data leakage on low-power sensors, but it instead focuses on message sizes rather than times.

Both FATS and STP show how attackers can use transmission times and device fingerprints to infer activities in smart homes [12, 101]. Along with AGE, these systems study communication side-channels. These approaches, however, target smart home devices that vary their transmission times. AGE instead protects low-power sensors that batch communication at regular intervals. Further, STP increases network traffic; this strategy is too costly for low-power devices. Finally, Das et al. use the communication volume from specific fitness trackers to infer activities [29]. Our work builds on this study by displaying how general-purpose adaptive policies leak information on multiple tasks. Further, we propose a novel framework to protect sensors in an energy-efficient manner.

**Website Fingerprinting** Website fingerprinting involves extracting information from encrypted network traffic [19–21, 68]. Previous work shows how packet sizes leak information about visited webpages [17, 49, 66, 83]. Dyer et al. show how coarse features are sufficient for fingerprinting [36]. Wright et al. shape traffic to standardize website patterns [117]. CRIME [34] and BREACH [45] use compressed sizes as a side-channel against TLS. Similar to these systems, we focus on a network traffic side-channel. Our work, however, concerns sensors with energy constraints; any bandwidth overhead leads to an untenable energy cost. We develop a defense that uses the numerical properties of sensor measurements to standardize communication with a negligible energy overhead.

**Low-Power Compression** Data compression is a common method to reduce the communication energy on low-power sensing devices. Mamaghanian et al. develop a compression algorithm to extend the lifetime of wearable body sensors [71]. Azar et al. apply lossy compression to IoT sensors and recover values using neural networks [13]. Other approaches use delta encoding [90], Chebyshev polynomials [110] and Huffman coding [72]. Similar to these systems, AGE can use lossy encoding to compress sensor messages. In contrast, our system creates fixed-length messages to avoid information leakage. Unlike standard compression, AGE will expand messages *on purpose* to meet the target size.

**Adaptive Sampling on Low-Power Devices** Adaptive sampling is a popular method for reducing energy consumption with minimal loss in error. Existing systems use adaptive procedures based on linear or autoregressive features [5, 7, 25, 63, 67, 96]. Further systems adapt the data collection pattern using Gaussian Processes [106], Reinforcement Learning [33, 78], and Neural Networks [22]. We show how three different adaptive policies leak information on multiple tasks. AGE protects these systems with minimal overhead.

# 7 DISCUSSION AND CONCLUSION

Adaptive Group Encoding (AGE) enforces data privacy by creating fixed-length messages using a lossy technique. We discuss the limitations of the proposed framework, as well as how AGE compares to alternative defenses.

**Batch Sizes** AGE offsets its slight computational overhead by reducing communication (§4.5). This strategy struggles when the target size is small. For example, if $M_B$ is 40 bytes, lowering the target by 30 bytes means AGE loses 75% of the message. When batches are small, padding provides a better defense as its overhead becomes minimal. In our evaluation, AGE uses at least 98 bytes, and we consider AGE superior to padding when the batches have at least 100 bytes.

**AGE and Energy Savings** AGE creates fixed-length messages that meet a target size $M_B$. This target size does not need to be derived from an energy constraint. Instead, AGE can work with any feasible target message size, and we use this property in the AGE framework to offset the encoding procedure's computational overhead (§4.5). Thus, similar to standard compression techniques, AGE supports the ability to save energy through reduced wireless communication. This feature allows AGE to display energy savings beyond those of the underlying adaptive sampling policy.

**Lossless Compression** AGE's creation of fixed-length messages makes it incompatible with lossless compression; applying compression after AGE will alter the final message size. Compression, however, is known to leak information about plaintext content [34, 45, 59]. Thus, even sensors using compression with non-adaptive sampling will suffer from data privacy issues. We leave solutions to compression's information leakage for future work.

**Alternative Defenses** We discuss two suboptimal alternative defenses. First, on-device inference [46] eliminates the communication side-channel by never transmitting raw values. However, it is difficult to understand and debug inference results without raw data; sending measurements provides flexibility during analysis. Additionally, real-world systems [111] process data at centralized servers. These applications should not suffer from data leakage.

A second alternative is to create same-sized messages by buffering excess values. This approach, however, increases the reporting latency, and this overhead worsens when the policy over-samples on consecutive batches. Further, this strategy needs enough memory—a limited resource on low-power sensors—to buffer excess values. Over-sampling for many consecutive batches would force the system to drop samples, producing a high error similar to the Pruned variant in §5.6.

**Adaptive Behavior and Information Leakage** In this work, we show that data-dependent adaptive sampling policies link their collection rates to sensed events. This relationship extends across multiple policies, leading to data leakage across a family of adaptive strategies. The scope of this problem hints at a larger issue with adaptive behavior. General adaptive frameworks based on data-dependent information tie their behavior to the source of their runtime feedback [9, 14, 23, 24, 27, 38, 47, 50–52, 58, 60, 76, 79, 86, 88, 97, 98, 108, 114, 120]. This property means the behavior of adaptive frameworks may show distinct relationships with input measurements. Thus, we conjecture that general adaptive frameworks suffer from information leakage in a manner similar to adaptive sampling.

We aim to explore this hypothesis in future work.

AGE presents a simple approach to protect adaptive sampling from leaking information through batched communication. The framework emphasizes having a low overhead in terms of both error and energy. Along with AGE's modular design, these properties make the framework compatible with existing sensor applications. We believe AGE represents a valuable step in bringing low-overhead security measures to resource-constrained sensing devices.

## ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful feedback to improve the final paper. We thank Tamara Lehman for shepherding this paper. This work was supported by NSF (grants CCF-2119184, CNS-1956180, CNS-1952050, CCF-1823032, CNS-1764039), ARO (grant W911NF1920321), and a DOE Early Career Award (grant DESC0014195 0003).

## A ARTIFACT APPENDIX

### A.1 Abstract

This artifact provides an implementation of Adaptive Group Encoding (AGE). AGE is a framework that protects adaptive sampling procedures on low-power sensors from leaking information through the size of batched messages. The system works by encoding all measurement batches as fixed-length messages, thereby breaking the relationship between the message size and the adaptive policy's collection rate. This repository implements AGE both in a simulated environment and on a microcontroller (MCU). The simulator, written in Python, represents the sensor and server as individual processes. These components communicate using a local (encrypted) socket, and the simulator tracks the sensor's energy consumption using traces from a TI MSP430 MCU. The hardware setting executes AGE on a TI MSP430 FR5994. The MCU transmits measurement batches to a separate server over a Bluetooth link. These experimental settings confirm AGE's ability to maintain the low error of adaptive sampling while preventing information leakage and incurring negligible energy overhead. The repository https://github.com/tejaskannan/adaptive-group-encoding contains all the code for this work.

### A.2 Artifact Check-List (Meta-Information)

- **Algorithm:** Adaptive Group Encoding (AGE)
- **Datasets:** Characters, EOG, Epilepsy, Human Activity Recognition, MNIST, Password, Pavement, Strawberry, Tiselac (Land Cover). All datasets are included in the artifact.
- **Run-time environment:** Python, C, Code Composer Studio
- **Hardware:** TI MSP430 FR5994, HM-10 BLE module, 4 jumper wires. This equipment is *not* needed for the simulator framework.
- **Metrics:** Mean Absolute Error (MAE) for sequence reconstruction, empirical Normalized Mutual Information (NMI), attacker inference accuracy
- **Experiments:** Data sampling using various policies, measuring reconstruction error, measuring mutual information, training and testing attack classifiers
- **How much disk space required?:** 1 GB for simulator. 4 GB if including pre-collected results for the MSP430 experiments.
- **How much time is needed to prepare workflow?:** Minimal setup for the simulator. About 1 hour for the MSP430 experiments.
- **How much time is needed to complete experiments (approximately)?:** 2 hours for end-to-end simulator experiments. About 10 minutes per MSP430 experiment.
- **Publicly available?:** Yes
- **Code licenses (if publicly available)?:** Apache License, Version 2.0
- **Archived (provide DOI)?:** 10.5281/zenodo.5747666

### A.3 Description

*A.3.1 How to access.* The code for this paper is contained in a GitHub repository [2]. The datasets, saved models, and pre-collected results are too large to fit in the GitHub repository. You may find this information in the project's associated Google Drive folder[3].

*A.3.2 Hardware dependencies.* The simulator requires a Linux computer with Python installed. The MCU experiments require a TI MSP430 FR5994 microcontroller, as well as an HM-10 BLE module and jumper wires. We include results from the MSP430 in the artifact for those without this equipment.

*A.3.3 Software dependencies.* The simulator framework is written in Python. We provide instructions for installing relevant software packages in the GitHub repository's README. The MSP430 implementation is written in C. We interface with the MSP430 using the TI Code Composer Studio (CCS) IDE. We have tested the implementation using CCSv10.1.0.

*A.3.4 Datasets.* We include all datasets in the Google Drive folder above.

*A.3.5 Models.* We include the trained sampling models in the Google Drive folder listed above.

### A.4 Installation

The README in the GitHub repository contains detailed instructions regarding installation.

### A.5 Experiment Workflow

The GitHub repository README contains instructions about how to execute experiments. Follow the README to reproduce the paper's results. To avoid the requirement for time consuming tasks, we include outputs from both the simulator and the TI MSP430. These result logs are found in the Google Drive folder linked above.

### A.6 Evaluation and Expected Results

The README file contains instructions about how to reproduce Tables 4-10 and Figures 5-7 in the paper.

## REFERENCES

[1] [n.d.]. *Graphical Password Dataset.*
[2] [n.d.]. TI MSP430 EnergyTrace Technology. https://www.ti.com/lit/ug/slau157as/slau157as.pdf?ts=1628556110549&ref_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FENERGYTRACE.
[3] [n.d.]. TI MSP430 FR5994 Datasheet. https://www.ti.com/lit/ds/symlink/msp430fr5994.pdf.

[2] https://github.com/tejaskannan/adaptive-group-encoding
[3] https://drive.google.com/drive/folders/1BrXn-Spc3GwbSmZu-xI5mLefBqNQ8vMa?usp=sharing

[4] Gunes Acar, Danny Yuxing Huang, Frank Li, Arvind Narayanan, and Nick Feamster. 2018. Web-based attacks to discover and control local IoT devices. In *Proceedings of the 2018 Workshop on IoT Security and Privacy*. 29–35.

[5] Cesare Alippi, Giuseppe Anastasi, Mario Di Francesco, and Manuel Roveri. 2009. An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors. *IEEE Transactions on Instrumentation and Measurement* 59, 2 (2009), 335–344.

[6] Cesare Alippi, Giuseppe Anastasi, Mario Di Francesco, and Manuel Roveri. 2009. Energy management in wireless sensor networks with energy-hungry sensors. *IEEE Instrumentation & Measurement Magazine* 12, 2 (2009), 16–23.

[7] Cesare Alippi, Giuseppe Anastasi, Cristian Galperti, Francesca Mancini, and Manuel Roveri. 2007. Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*. IEEE, 1–6.

[8] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. 2012. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*. Springer, 216–223.

[9] Jason Ansel, Maciej Pacula, Yee Lok Wong, Cy Chan, Marek Olszewski, Una-May O'Reilly, and Saman Amarasinghe. 2012. Siblingrivalry: Online Autotuning Through Local Competitions. In *Proceedings of the 2012 International Conference on Compilers, Architectures and Synthesis for Embedded Systems* (Tampere, Finland) *(CASES '12)*. ACM, New York, NY, USA, 91–100. https://doi.org/10.1145/2380403.2380425

[10] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. 2017. Understanding the mirai botnet. In *26th {USENIX} security symposium ({USENIX} Security 17)*. 1093–1110.

[11] Geoff Appelboom, Elvis Camacho, Mickey E Abraham, Samuel S Bruce, Emmanuel LP Dumont, Brad E Zacharia, Randy D'Amico, Justin Slomian, Jean Yves Reginster, Olivier Bruyère, et al. 2014. Smart wearable body sensors for patient self-assessment and monitoring. *Archives of public health* 72, 1 (2014), 1–9.

[12] Noah Apthorpe, Danny Yuxing Huang, Dillon Reisman, Arvind Narayanan, and Nick Feamster. 2019. Keeping the smart home private with smart(er) IoT traffic shaping. *Proceedings on Privacy Enhancing Technologies* 2019, 3 (2019), 128–148.

[13] Joseph Azar, Abdallah Makhoul, Mahmoud Barhamgi, and Raphaël Couturier. 2019. An energy efficient IoT data compression approach for edge machine learning. *Future Generation Computer Systems* 96 (2019), 168–175.

[14] Woongki Baek and Trishul M. Chilimbi. 2010. Green: A Framework for Supporting Energy-conscious Programming Using Controlled Approximation. In *Proceedings of the 31st ACM Conference on Programming Language Design and Implementation* (Toronto, Ontario, Canada) *(PLDI '10)*. ACM, New York, NY, USA, 198–209. https://doi.org/10.1145/1806596.1806620

[15] Ravi Bagree, Vishwas Raj Jain, Aman Kumar, and Prabhat Ranjan. 2010. Tigercense: Wireless image sensor network to monitor tiger movement. In *International Workshop on Real-world Wireless Sensor Networks*. Springer, 13–24.

[16] Bruhadeshwar Bezawada, Maalvika Bachani, Jordan Peterson, Hossein Shirazi, Indrakshi Ray, and Indrajit Ray. 2018. Behavioral fingerprinting of IoT devices. In *Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security*. 41–50.

[17] Sanjit Bhat, David Lu, Albert Kwon, and Srinivas Devadas. 2019. Var-CNN: A data-efficient website fingerprinting attack based on deep learning. *Proceedings on Privacy Enhancing Technologies* 4 (2019), 292–310.

[18] Sergey Bratus, Cory Cornelius, David Kotz, and Daniel Peebles. 2008. Active behavioral fingerprinting of wireless devices. In *Proceedings of the first ACM conference on Wireless network security*. 56–61.

[19] Xiang Cai, Rishab Nithyanand, and Rob Johnson. 2014. CS-BuFLO: A congestion sensitive website fingerprinting defense. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*. 121–130.

[20] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. 2014. A systematic approach to developing and evaluating website fingerprinting defenses. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 227–238.

[21] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. 2012. Touching from a distance: Website fingerprinting attacks and defenses. In *Proceedings of the 2012 ACM conference on Computer and communications security*. 605–616.

[22] Víctor Campos, Brendan Jou, Xavier Giró-i Nieto, Jordi Torres, and Shih-Fu Chang. 2017. Skip RNN: Learning to skip state updates in recurrent neural networks. *arXiv preprint arXiv:1708.06834* (2017).

[23] Anthony Canino and Yu David Liu. 2017. Proactive and Adaptive Energy-aware Programming with Mixed Typechecking. In *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation* (Barcelona, Spain) *(PLDI 2017)*. ACM, New York, NY, USA, 217–232. https://doi.org/10.1145/3062341.3062356

[24] Anthony Canino, Yu David Liu, and Hidehiko Masuhara. 2018. Stochastic Energy Optimization for Mobile GPS Applications. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Lake Buena Vista, FL, USA) *(ESEC/FSE 2018)*. ACM, New York, NY, USA, 703–713. https://doi.org/10.1145/3236024.3236076

[25] Supriyo Chatterjea and Paul Havinga. 2008. An adaptive and autonomous sensor sampling frequency control scheme for energy-efficient data acquisition in wireless sensor networks. In *International Conference on Distributed Computing in Sensor Systems*. Springer, 60–78.

[26] Konstantinos Chatzikokolakis, Tom Chothia, and Apratim Guha. 2010. Statistical measurement of information leakage. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 390–404.

[27] H. Chen, M. Song, J. Song, A. Gavrilovska, and K. Schwan. 2011. HEaRS: A Hierarchical Energy-Aware Resource Scheduler for Virtualized Data Centers. In *2011 IEEE International Conference on Cluster Computing*. 508–512. https://doi.org/10.1109/CLUSTER.2011.60

[28] Hari Cherupalli, Henry Duwe, Weidong Ye, Rakesh Kumar, and John Sartori. 2017. Software-based gate-level information flow security for IoT systems. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*. 328–340.

[29] Aveek K Das, Parth H Pathak, Chen-Nee Chuah, and Prasant Mohapatra. 2016. Uncovering privacy leakage in BLE network traffic of wearable fitness trackers. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*. 99–104.

[30] Bradley Denby and Brandon Lucia. 2019. Orbital edge computing: Machine inference in space. *IEEE Computer Architecture Letters* 18, 1 (2019), 59–62.

[31] Bradley Denby and Brandon Lucia. 2020. Orbital edge computing: Nanosatellite constellations as a new class of computer system. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*. 939–954.

[32] Amol Deshpande, Carlos Guestrin, Samuel R Madden, Joseph M Hellerstein, and Wei Hong. 2004. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. 588–599.

[33] Gabriel Martins Dias, Maddalena Nurchis, and Boris Bellalta. 2016. Adapting sampling interval of sensor networks using on-line reinforcement learning. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. IEEE, 460–465.

[34] Thai Duong and Julianno Rizzo. [n.d.]. The CRIME attack. https://docs.google.com/presentation/d/11eBmGiHbYcHR9gL5nDyZChu_-lCa2GizeuOfaLU2HOU.

[35] Morris Dworkin, Elaine Barker, James Nechvatal, James Foti, Lawrence Bassham, E. Roback, and James Dray. 2001. Advanced Encryption Standard (AES).

[36] Kevin P Dyer, Scott E Coull, Thomas Ristenpart, and Thomas Shrimpton. 2012. Peek-a-boo, I still see you: Why efficient traffic analysis countermeasures fail. In *2012 IEEE symposium on security and privacy*. IEEE, 332–346.

[37] Fuming Fang and Takahiro Shinozaki. 2018. Electrooculography-based continuous eye-writing recognition system for efficient assistive communication systems. *PloS one* 13, 2 (2018), e0192684.

[38] Jason Flinn and M. Satyanarayanan. 1999. Energy-Aware Adaptation for Mobile Applications. In *Proceedings of the Seventeenth ACM Symposium on Operating Systems Principles* (Charleston, South Carolina, USA) *(SOSP '99)*. Association for Computing Machinery, New York, NY, USA, 48–63. https://doi.org/10.1145/319151.319155

[39] David Formby, Preethi Srinivasan, Andrew M Leonard, Jonathan D Rogers, and Raheem A Beyah. 2016. Who's in control of your control system? Device fingerprinting for cyber-physical systems. In *NDSS*.

[40] Yoav Freund, Robert Schapire, and Naoki Abe. 1999. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence* 14, 771-780 (1999), 1612.

[41] Warren Frick and Carlos Niederstrasser. 2018. Small launch vehicles-a 2018 state of the industry survey. (2018).

[42] Ke Gao, Cherita Corbett, and Raheem Beyah. 2010. A passive approach to wireless device fingerprinting. In *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*. IEEE, 383–392.

[43] Bugra Gedik, Ling Liu, and S Yu Philip. 2007. ASAP: An adaptive sampling approach to data collection in sensor networks. *IEEE Transactions on Parallel and distributed systems* 18, 12 (2007), 1766–1783.

[44] Giacomo Giuliari, Tommaso Ciussani, Adrian Perrig, and Ankit Singla. 2021. ICARUS: Attacking low Earth orbit satellite networks. In *2021 {USENIX} Annual Technical Conference ({USENIX} {ATC} 21)*. 317–331.

[45] Yoel Gluck, Neal Harris, and Angelo Prado. 2013. BREACH: reviving the CRIME attack. *Unpublished manuscript* (2013).

[46] Graham Gobieski, Brandon Lucia, and Nathan Beckmann. 2019. Intelligence beyond the edge: Inference on intermittent embedded systems. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. 199–213.

[47] Ashvin Goel, David Steere, Calton Pu, and Jonathan Walpole. 1998. *SWiFT: A Feedback Control and Dynamic Reconfiguration Toolkit*. Technical Report.

[48] Phillip Good. 2013. *Permutation tests: A practical guide to resampling methods for testing hypotheses*. Springer Science & Business Media.

[49] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. 2009. Website fingerprinting: Attacking popular privacy enhancing technologies with the

multinomial naïve-bayes classifier. In *Proceedings of the 2009 ACM workshop on Cloud computing security*. 31–42.

[50] Henry Hoffmann. 2015. JouleGuard: Energy Guarantees for Approximate Applications. In *Proceedings of the 25th Symposium on Operating Systems Principles* (Monterey, California) *(SOSP '15)*. ACM, New York, NY, USA, 198–214. https://doi.org/10.1145/2815400.2815403

[51] Henry Hoffmann, Jim Holt, George Kurian, Eric Lau, Martina Maggio, Jason E. Miller, Sabrina M. Neuman, Mahmut Sinangil, Yildiz Sinangil, Anant Agarwal, Anantha P. Chandrakasan, and Srinivas Devadas. 2012. Self-Aware Computing in the Angstrom Processor. In *Proceedings of the 49th Annual Design Automation Conference* (San Francisco, California) *(DAC '12)*. Association for Computing Machinery, New York, NY, USA, 259–264. https://doi.org/10.1145/2228360.2228409

[52] Henry Hoffmann, Axel Jantsch, and Nikil D. Dutt. 2020. Embodied Self-Aware Computing Systems. *Proc. IEEE* 108, 7 (2020), 1027–1046. https://doi.org/10.1109/JPROC.2020.2977054

[53] JK Holland, EK Kemsley, and RH Wilson. 1998. Use of Fourier transform infrared spectroscopy and partial least squares regression for the detection of adulteration of strawberry purees. *Journal of the Science of Food and Agriculture* 76, 2 (1998), 263–269.

[54] Kevin Hung, Yuan-Ting Zhang, and B Tai. 2004. Wearable medical devices for tele-home healthcare. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Vol. 2. IEEE, 5384–5387.

[55] Dino Ienco, Raffaele Gaetano, Claire Dupaquier, and Pierre Maurel. 2017. Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geoscience and Remote Sensing Letters* 14, 10 (2017), 1685–1689.

[56] Pandurang Kamat, Wenyuan Xu, Wade Trappe, and Yanyong Zhang. 2007. Temporal privacy in wireless sensor networks. In *27th International Conference on Distributed Computing Systems (ICDCS'07)*. IEEE, 23–23.

[57] Tejas Kannan and Henry Hoffmann. 2021. Budget RNNs: Multi-capacity neural networks to improve in-sensor inference under energy budgets. In *2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 143–156.

[58] Aman Kansal, Scott Saponas, A.J. Bernheim Brush, Kathryn S. McKinley, Todd Mytkowicz, and Ryder Ziola. 2013. The Latency, Accuracy, and Battery (LAB) Abstraction: Programmer Productivity and Energy Efficiency for Continuous Mobile Context Sensing. In *Proceedings of the 2013 ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications* (Indianapolis, Indiana, USA) *(OOPSLA '13)*. ACM, New York, NY, USA, 661–676. https://doi.org/10.1145/2509136.2509541

[59] John Kelsey. 2002. Compression and information leakage of plaintext. In *International Workshop on Fast Software Encryption*. Springer, 263–276.

[60] Minyoung Kim, Mark-Oliver Stehr, Carolyn Talcott, Nikil Dutt, and Nalini Venkatasubramanian. 2013. XTune: A Formal Methodology for Cross-Layer Tuning of Mobile Embedded Systems. *ACM Trans. Embed. Comput. Syst.* 11, 4, Article 73 (Jan. 2013), 23 pages. https://doi.org/10.1145/2362336.2362340

[61] Paul Kocher, Joshua Jaffe, Benjamin Jun, and Pankaj Rohatgi. 2011. Introduction to differential power analysis. *Journal of Cryptographic Engineering* 1, 1 (2011), 5–27.

[62] Tarald O Kvålseth. 2017. On normalized mutual information: Measure derivations and properties. *Entropy* 19, 11 (2017), 631.

[63] Yee Wei Law, Supriyo Chatterjea, Jiong Jin, Thomas Hanselmann, and Marimuthu Palaniswami. 2009. Energy-efficient data acquisition by adaptive sampling for wireless sensor networks. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*. 1146–1151.

[64] Yann LeCun. 1998. The MNIST database of handwritten digits. *http://yann.lecun. com/exdb/mnist/* (1998).

[65] Patrick Leu, Ivan Puddu, Aanjhan Ranganathan, and Srdjan Čapkun. 2018. I send, therefore I leak: Information leakage in low-power wide area networks. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 23–33.

[66] Marc Liberatore and Brian Neil Levine. 2006. Inferring the source of encrypted HTTP connections. In *Proceedings of the 13th ACM conference on Computer and communications security*. 255–263.

[67] Ping Lou, Liang Shi, Xiaomei Zhang, Zheng Xiao, and Junwei Yan. 2020. A data-driven adaptive sampling method based on edge computing. *Sensors* 20, 8 (2020), 2174.

[68] Liming Lu, Ee-Chien Chang, and Mun Choon Chan. 2010. Website fingerprinting and identification using ordered feature sequences. In *European Symposium on Research in Computer Security*. Springer, 199–214.

[69] Matt Mahoney. 2012. *Data compression explained*. mattmahoney.net, Chapter 1,1.

[70] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. 2002. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. 88–97.

[71] Hossein Mamaghanian, Nadia Khaled, David Atienza, and Pierre Vandergheynst. 2011. Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes. *IEEE Transactions on Biomedical Engineering* 58, 9 (2011), 2456–2466.

[72] Francesco Marcelloni and Massimo Vecchio. 2008. A simple algorithm for data compression in wireless sensor networks. *IEEE communications letters* 12, 6 (2008), 411–413.

[73] Gaurav Mathur, Peter Desnoyers, Paul Chukiu, Deepak Ganesan, and Prashant Shenoy. 2009. Ultra-low power data storage for sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 5, 4 (2009), 1–34.

[74] Gaurav Mathur, Peter Desnoyers, Deepak Ganesan, and Prashant Shenoy. 2006. Ultra-low power data storage for sensor networks. In *Proceedings of the 5th International Conference on IPSN/SPOTS*.

[75] Thomas S Messerges, Ezzat A Dabbish, and Robert H Sloan. 2002. Examining smart-card security under the threat of power analysis attacks. *IEEE transactions on computers* 51, 5 (2002), 541–552.

[76] Nikita Mishra, Connor Imes, John D Lafferty, and Henry Hoffmann. 2018. CALOREE: Learning control for predictable latency and low energy. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*. 184–198.

[77] Hooman Mohajeri Moghaddam, Gunes Acar, Ben Burgess, Arunesh Mathur, Danny Yuxing Huang, Nick Feamster, Edward W Felten, Prateek Mittal, and Arvind Narayanan. 2019. Watching you watch: The tracking ecosystem of over-the-top TV streaming devices. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 131–147.

[78] Abdulmajid Murad, Frank Alexander Kraemer, Kerstin Bach, and Gavin Taylor. 2020. Information-driven adaptive sensing based on deep reinforcement learning. In *Proceedings of the 10th International Conference on the Internet of Things*. 1–8.

[79] Mischa Möstl, Johannes Schlatow, Rolf Ernst, Henry Hoffmann, Arif Merchant, and Alexander Shraer. 2016. Self-aware systems for the Internet-of-Things. In *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*. 1–9.

[80] Leslie M Naylor, Michael J Wisdom, and Robert G Anthony. 2009. Behavioral responses of North American elk to recreational activity. *The Journal of Wildlife Management* 73, 3 (2009), 328–338.

[81] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. 2016. Phased LSTM: Accelerating recurrent network training for long or event-based sequences. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 3889–3897.

[82] Markus Ojala and Gemma C Garriga. 2010. Permutation tests for studying classifier performance. *Journal of Machine Learning Research* 11, 6 (2010).

[83] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. 2016. Website fingerprinting at internet scale. In *NDSS*.

[84] Jeffrey Pang, Ben Greenstein, Ramakrishna Gummadi, Srinivasan Seshan, and David Wetherall. 2007. 802.11 user fingerprinting. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*. 99–110.

[85] Armen Poghosyan and Alessandro Golkar. 2017. CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions. *Progress in Aerospace Sciences* 88 (2017), 59–83.

[86] Raghavendra Pradyumna Pothukuchi, Sweta Yamini Pothukuchi, Petros Voulgaris, and Josep Torrellas. 2018. Yukta: Multilayer Resource Controllers to Maximize Efficiency. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. 505–518. https://doi.org/10.1109/ISCA.2018.00049

[87] Jordi Puig-Suari, Clark Turner, and William Ahlgren. 2001. Development of the standard CubeSat deployer and a CubeSat class PicoSatellite. In *2001 IEEE aerospace conference proceedings (cat. No. 01TH8542)*, Vol. 1. IEEE, 1–347.

[88] Amir M. Rahmani, Bryan Donyanavard, Tiago Mück, Kasra Moazzemi, Axel Jantsch, Onur Mutlu, and Nikil Dutt. 2018. SPECTR: Formal Supervisory Control and Coordination for Many-Core Systems Resource Management. *SIGPLAN Not.* 53, 2 (March 2018), 169–183. https://doi.org/10.1145/3296957.3173199

[89] Kasper Bonne Rasmussen and Srdjan Capkun. 2007. Implications of radio fingerprinting on the security of sensor networks. In *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops-SecureComm 2007*. IEEE, 331–340.

[90] Abdeldjalil Saidani, Xiang Jianwen, and Deloula Mansouri. 2020. A Lossless Compression Approach Based on Delta Encoding and T-RLE in WSNs. *Wireless Communications and Mobile Computing* 2020 (2020).

[91] Robert E Schapire. 2013. Explaining AdaBoost. In *Empirical inference*. Springer, 37–52.

[92] M Scheffler and E Hirt. 2005. Wearable devices for telemedicine applications. *Journal of telemedicine and telecare* 11, 1_suppl (2005), 11–14.

[93] Nader Sehatbakhsh, Alireza Nazari, Haider Khan, Alenka Zajic, and Milos Prvulovic. 2019. Emma: Hardware/software attestation framework for embedded systems using electromagnetic signals. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. 983–995.

[94] Jayaprakash Selvaraj, Gökçen Yılmaz Dayanıklı, Neelam Prabhu Gaunkar, David Ware, Ryan M Gerdes, and Mani Mina. 2018. Electromagnetic induction attacks

against embedded systems. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. 499–510.

[95] Sandra Siby, Rajib Ranjan Maiti, and Nils Tippenhauer. 2017. IoTscanner: Detecting and classifying privacy threats in IoT neighborhoods. *arXiv preprint arXiv:1701.05007* (2017).

[96] João Marco C Silva, Kalil Araujo Bispo, Paulo Carvalho, and Solange Rito Lima. 2017. LiteSense: An adaptive sensing scheme for WSNs. In *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 1209–1212.

[97] David C. Snowdon, Etienne Le Sueur, Stefan M. Petters, and Gernot Heiser. 2009. Koala: A Platform for OS-Level Power Management. In *Proceedings of the 4th ACM European Conference on Computer Systems* (Nuremberg, Germany) *(EuroSys '09)*. Association for Computing Machinery, New York, NY, USA, 289–302. https://doi.org/10.1145/1519065.1519097

[98] Jacob Sorber, Alexander Kostadinov, Matthew Garber, Matthew Brennan, Mark D. Corner, and Emery D. Berger. 2007. Eon: A Language and Runtime System for Perpetual Systems. In *In Proceedings of The Fifth International ACM Conference on Embedded Networked Sensor Systems (SenSys '07), Syndey*.

[99] Jacob Sorber, Minho Shin, Ronald Peterson, Cory Cornelius, Shrirang Mare, Aarathi Prasad, Zachary Marois, Emma Smithayer, and David Kotz. 2012. An amulet for trustworthy wearable mHealth. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*. 1–6.

[100] Vinicius MA Souza. 2018. Asphalt pavement classification using smartphone accelerometer and complexity invariant distance. *Engineering Applications of Artificial Intelligence* 74 (2018), 198–211.

[101] Vijay Srinivasan, John Stankovic, and Kamin Whitehouse. 2008. Protecting your daily in-home activity information from a wireless snooping attack. In *Proceedings of the 10th international conference on Ubiquitous computing*. 202–211.

[102] Frank Stajano and Ross Anderson. 1999. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *International workshop on security protocols*. Springer, 172–182.

[103] John A. Stankovic. 1988. Misconceptions about real-time computing: A serious problem for next-generation systems. *Computer* 21, 10 (1988), 10–19.

[104] Jung-Woong Sung and Seung-Jae Han. 2017. Data bundling for energy efficient communication of wearable devices. *Computer Networks* 121 (2017), 76–88.

[105] Sofiane Takarabt, Alexander Schaub, Adrien Facon, Sylvain Guilley, Laurent Sauvage, Youssef Souissi, and Yves Mathieu. 2019. Cache-timing attacks still threaten IoT devices. In *International Conference on Codes, Cryptology, and Information Security*. Springer, 13–30.

[106] Yew Teck Tan, Abhinav Kunapareddy, and Marin Kobilarov. 2018. Gaussian process adaptive sampling using the cross-entropy method for environmental sensing and monitoring. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 6220–6227.

[107] Armen Toorian, Ken Diaz, and Simon Lee. 2008. The cubesat approach to space access. In *2008 IEEE Aerospace Conference*. IEEE, 1–14.

[108] Konstantinos Tovletoglou, Lev Mukhanov, Dimitrios S. Nikolopoulos, and Georgios Karakonstantis. 2020. HaRMony: Heterogeneous-Reliability Memory and QoS-Aware Energy Management on Virtualized Servers. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems* (Lausanne, Switzerland) *(ASPLOS '20)*. Association for Computing Machinery, New York, NY, USA, 575–590. https://doi.org/10.1145/3373376.3378489

[109] Wade Trappe, Richard Howard, and Robert S Moore. 2015. Low-energy security: Limits and opportunities in the internet of things. *IEEE Security & Privacy* 13, 1 (2015), 14–21.

[110] Arijit Ukil, Soma Bandyopadhyay, Aniruddha Sinha, and Arpan Pal. 2015. Adaptive sensor data compression in IoT systems: Sensor data analytics based approach. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5515–5519.

[111] Deepak Vasisht, Zerina Kapetanovic, Jongho Won, Xinxin Jin, Ranveer Chandra, Sudipta Sinha, Ashish Kapoor, Madhusudhan Sudarshan, and Sean Stratman. 2017. Farmbeats: An iIoT platform for data-driven agriculture. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*. 515–529.

[112] Jose R Villar, Paula Vergara, Manuel Menéndez, Enrique de la Cal, Víctor M González, and Javier Sedano. 2016. Generalized models for the classification of abnormal movements in daily life and its applicability to epilepsy convulsion recognition. *International journal of neural systems* 26, 06 (2016).

[113] He Wang, Ted Tsung-Te Lai, and Romit Roy Choudhury. 2015. Mole: Motion leaks through smartwatch sensors. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 155–166.

[114] Shu Wang, Chi Li, Henry Hoffmann, Shan Lu, William Sentosa, and Achmad Imam Kistijantoro. 2018. Understanding and Auto-Adjusting Performance-Sensitive Configurations. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems* (Williamsburg, VA, USA) *(ASPLOS '18)*. ACM, New York, NY, USA, 154–168. https://doi.org/10.1145/3173162.3173206

[115] Rebecca Willett, Aline Martin, and Robert Nowak. 2004. Backcasting: Adaptive sampling for sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*. 124–133.

[116] Ben H Williams, Marc Toussaint, and Amos J Storkey. 2006. Extracting motion primitives from natural handwriting data. In *International Conference on Artificial Neural Networks*. Springer, 634–643.

[117] Charles V Wright, Scott E Coull, and Fabian Monrose. 2009. Traffic morphing: An efficient defense against statistical traffic analysis. In *NDSS*, Vol. 9. Citeseer.

[118] Pei Zhang and Margaret Martonosi. 2008. Locale: Collaborative localization estimation for sparse mobile sensor networks. In *2008 International Conference on Information Processing in Sensor Networks (IPSN 2008)*. IEEE, 195–206.

[119] Pei Zhang, Christopher M Sadler, Stephen A Lyon, and Margaret Martonosi. 2004. Hardware design experiences in ZebraNet. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*. 227–238.

[120] Ronghua Zhang, Chenyang Lu, Tarek F. Abdelzaher, and John A. Stankovic. 2002. ControlWare: a middleware architecture for feedback control of software performance. In *Proceedings 22nd International Conference on Distributed Computing Systems*. 301–310. https://doi.org/10.1109/ICDCS.2002.1022267