Algorithmic Data Minimization for Machine Learning over Internet-of-Things Data Streams

Ted Shaowang
The University of Chicago
swjz@uchicago.edu

Shinan Liu
The University of Hong Kong
shinan6@hku.hk

Jonatas Marques
The University of Chicago
jmarques@uchicago.edu

Nick Feamster
The University of Chicago
feamster@uchicago.edu

Sanjay Krishnan
The University of Chicago
skr@uchicago.edu

ABSTRACT

Machine learning can analyze vast amounts of data generated by IoT devices to identify patterns, make predictions, and enable real-time decision-making. This raises significant privacy concerns, necessitating the application of data minimization – a foundational principle in emerging data regulations, which mandates that service providers only collect data that is directly relevant and necessary for a specified purpose. Despite its importance, data minimization lacks a precise technical definition in the context of sensor data, where collections of *weak signals* make it challenging to apply a binary "relevant and necessary" rule. This paper provides a technical interpretation of data minimization in the context of sensor streams, explores practical methods for implementation, and addresses the challenges involved. Through our approach, we demonstrate that our framework can reduce user identifiability by up to 16.7% while maintaining accuracy loss below 1%, offering a viable path toward privacy-preserving IoT data processing.

PVLDB Reference Format:
Ted Shaowang, Shinan Liu, Jonatas Marques, Nick Feamster, and Sanjay Krishnan. Algorithmic Data Minimization for Machine Learning over Internet-of-Things Data Streams. PVLDB, 18(13): 5652 - 5661, 2025.
doi:10.14778/3773731.3773740

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/swjz/data-minimization.

1 INTRODUCTION

Internet of Things (IoT) systems consist of interconnected devices that collect and transmit data through embedded sensors, enabling smart functionalities across various domains. Machine learning plays a critical role in analyzing this sensor data to unlock valuable insights and optimize operations [13, 17–19, 42]. For example, in smart homes, machine learning models can predict energy consumption patterns to optimize heating or cooling schedules, reducing costs and environmental impact. In healthcare, IoT devices, such as wearable sensors, can track vital signs and alert medical professionals to anomalies, enabling proactive care. These applications highlight the vast opportunities IoT and machine learning offer [14, 38], from enhancing personalized experiences to improving efficiency and fostering innovation across industries.

However, with these opportunities come certain acute risks. IoT data, e.g., from smart home systems, can inadvertently leak identifiable information due to vulnerabilities in data handling and transmission [6, 28, 39]. Devices such as smart speakers, thermostats, and security cameras collect and transmit vast amounts of personal data, including voice commands, daily routines, and even video footage. Such risks highlight the critical need for robust measures and transparency in IoT ecosystems to protect user data and privacy. Thus, every service provider building an intelligent IoT system must tradeoff the utility of collecting data for prediction versus the potentially identifiable characteristics manifested in the data [15, 44].

To help navigate such tradeoffs, emerging frameworks like the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) have introduced new principles for managing user data [3, 7]. Among these, *data minimization* stands out as a foundational principle. It mandates that service providers collect only the user data that is directly "relevant and necessary" to achieve a specified purpose [8, 15, 34, 36]. The principle of data minimization is important because it reduces the risk of data breaches, ensures compliance with privacy regulations, and limits the collection of unnecessary or excessive personal information, thereby protecting individual privacy and fostering trust [34].

Unfortunately, a clear technical definition of data minimization is missing in the context of IoT analytics. In a sensor setting, a threshold of "relevant and necessary" may be hard to characterize as every data facet may be correlated in some way to the desired prediction target. In these settings, we often have collections of weak signals that contribute both towards a prediction target and towards re-identifying individuals. Determining what data is truly necessary remains an ill-posed task.

In this work, we address this challenge by proposing a novel approach to data minimization. Our method focuses on selectively discarding features that, while potentially useful for identifying individual users, offer minimal contribution to the task at hand. This strategy aims to strike a balance between privacy preservation and task accuracy, ensuring that privacy-sensitive information is minimized without significantly compromising system performance. It is crucial to clarify that data minimization does not equate to simply minimizing the size of the collected dataset (i.e., rows). We focus on

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 18, No. 13 ISSN 2150-8097.
doi:10.14778/3773731.3773740

feature minimization to address data leakage during inference time. Rather, the goal is to minimize *identifiability* of users inferred from the dataset based on the features used for a machine learning task.

This paper makes the following contributions. (1) A formal model for data minimization based on a two-player game in which a model provider tries to maximize accuracy while an adversary tries to maximize identifiability of users. (2) We present practical algorithms to identify effective provider strategies as solutions to this game, whose optimal solution is computationally intractable. (3) We show experimental results across 7 IoT datasets. While no single heuristic is universally effective, we show that data minimization strategies that *do not* model this two-player game with an adversary are generally less effective. (4) We provide concrete recommendations on best practices to improve the effectiveness of data minimization in real-world applications.

2 BACKGROUND

Machine learning over multimodal IoT data involves integrating information from diverse sensor sources—such as cameras, accelerometers, and biosensors—to improve decision-making, prediction accuracy, and situational awareness. These sensors often capture complementary aspects of an environment, enabling more robust models for applications like autonomous vehicles, healthcare monitoring, and industrial automation. For example, in medical diagnostics, combining EEG (brain activity) and ECG (heart activity) data can enhance early detection of neurological and cardiovascular disorders. Multimodal sensing is characterized by fusing information from multiple "weak" signals – rather than a single strong one like in classical AI problems such as NLP and Computer Vision. Sensor data is plagued with missing or degraded signals, which can be mitigated through multiple sensing views of the same phenomenon.

Unfortunately, the nature of IoT applications means that the data collected could leak identifying information. Consider the EEG application above. While clinically useful to share EEG datasets, these data can inadvertently reveal sensitive information. Idiosyncrasies in sensing hardware or the individual could leak data to an adversary. This issue is particularly pronounced in cases where specific signals are unique to certain users. To address this, privacy-preserving frameworks are needed to help data and model providers navigate accuracy vs. identifiability tradeoffs.

2.1 Formal Approaches to Data Minimization

The principle of data minimization is a fundamental concept in data privacy and security that emphasizes collecting, processing, and storing only the minimum amount of personal data necessary to achieve a specific purpose. This principle is a core requirement in many data protection regulations, such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). By limiting data collection, organizations reduce the risk of data breaches, unauthorized access, and misuse while also ensuring compliance with legal and ethical standards.

Unfortunately, the IoT domain introduces specific challenges. It isn't a clear yes/no question to whether a particular signal is absolutely needed for a task or is overly identifying. Every signal has a degree of utility as well as a degree of identifiability. This degree of utility can be highly dynamic, varying across the feature space and over users. This paper explores formal approaches to data minimization in such applications.

Data minimization can take various forms, including the removal of data points (rows) and features (columns). Removing rows as a form of data minimization is highly valuable for ensuring training datasets do not contain too much private information. Conversely, feature-minimization largely helps with reducing the amount of data given to an inference service [37]. For the purpose of this paper, we focus exclusively on removing unnecessary features (columns) for optimizing the inference problem. Both approaches are complementary as they address different stages of the machine-learning pipeline. This approach is comparable to traditional feature selection methods, as discussed in §2.2.2. However, traditional methods typically prioritize utility by selecting features that maximize model accuracy. They overlook the privacy implications of certain features, which may inadvertently reveal user identity. We argue that an effective data minimization method should strike a balance between utility and privacy [35]. It should retain features that are essential for the predictive task while removing those that reveal user information, provided their contribution to accuracy is minimal.

A key gap in existing work is the lack of a clear definition of user identifiability – what does it mean for a set of features to reveal user information? We propose a practical definition: user identifiability can be measured by training a user-classification model on such features. A privacy-preserving set of features should result in low accuracy for this model, indicating that user information cannot be inferred from these features.

Definition 1 (Data Minimization)**.** The goal of data (feature) minimization is to find a subset S of feature set F ($S \subseteq F$):

$$\min \text{Identifiability}[t(S)]$$

subject to:

$$\text{Accuracy}[p(S)] \geq (1 - l) \times \text{Accuracy}[p(F)]$$

where p is the primary model for the predictive task, and t is a user-classification model. Identifiability is defined as the classification accuracy of t.

When certain features strongly influence both predictive accuracy and user identifiability, the tradeoff between utility and privacy becomes more nuanced. The extent to which users are willing to sacrifice predictive performance for enhanced privacy is controlled by the tunable tolerance parameter l. This threshold plays a critical role in practice, as users may be unwilling to compromise accuracy arbitrarily in favor of privacy.

2.2 Related Work and Baselines

Existing work mostly approaches privacy-aware data management problems from one of the three perspectives: (1) selection strategy, (2) privacy mechanism, and (3) computation location. Privacy mechanisms include differential privacy [1], homomorphic encryption [25] and secure multi-party computation [5]. Computation locations include centralized, federated learning [26] and edge-based serving [30–32]. In this section, we mainly focus on selection strategies as they are more relevant to our work.

2.2.1 Differential privacy. Differential privacy is a well-studied technique to anonymize private data by injecting calibrated noise to a dataset while preserving its overall statistics [4]. This method ensures that individual data cannot be identified by a statistical analysis or release. While differential privacy makes it harder to reveal information about a specific user, it does not take into consideration what data is relevant and necessary for the intended task. This limitation can lead to the retention of data that is irrelevant to the task and accuracy loss.

2.2.2 *Feature selection.*

Existing feature selection methods focus on selecting a subset of features that are most significant to the model's performance. Although this process does reduce the total amount of information, it overlooks the possibility that the most predictive features might also be the most privacy-intrusive. Consequently, these methods may inadvertently retain features that pose a higher risk to user identifiability, despite their contribution to the model's accuracy.

Feature hashing. The hashing trick [24, 40] can be used as a tool for feature selection as it hashes a large number of features into a smaller number of indices. A fixed-size output dimensionality is guaranteed, but hash collisions can occur and add noise to data.

PCA-based methods. PCA [27] and Sparse PCA [45] are examples of linear methods to reduce the dimensionality of features. For dense datasets, PCA would work just fine but Sparse PCA works better when the dataset is high-dimensional and sparse.

2.2.3 *Feature scoring.*

Another line of work discusses how to attribute scores to each feature so it is easier to determine the contributions made by each feature.

Entropy-based methods. Mutual information is a measure of shared information between two variables, or the entropy lost by knowing one of the variables. [16]. A high mutual information score means that knowing X reduces more uncertainty about Y. In the context of feature selection, we want to keep the features that share high mutual information with labels.

Following ideas from [35], we apply mutual information as a metric for feature *utility score*, and the entropy difference between including the feature vs. excluding the feature as a metric for feature *privacy score*. This is just the conditional entropy of the included feature given all others, i.e., how identifying to an arbitrary labeling is this feature above what is already in the dataset. The normalized sum of utility score and privacy score can be used as a *privacy-utility tradeoff score*. However, we find that high entropy features do not necessarily mean privacy-invasive in many cases, as long as the additional entropy does not correlate with personal identity.

SHAP-based methods. SHAP (SHapley Additive exPlanations) [21] is a game-theoretic approach to explain machine learning models by attributing changes in model outputs to each contributing feature based on Shapley values.

Impurity-based feature importance. Gini impurity is yet another metric for feature selection, particularly in tree-based models such as random forests and gradient boosted trees [10]. This method evaluates the importance of a feature based on its contribution to reducing impurity in the decision nodes of the model. Features that lead to greater reductions in impurity across multiple splits are considered more informative and receive higher importance scores.

3 FORMAL MODEL FOR DATA MINIMIZATION OVER WEAK SIGNALS

Next, we present a method for operationalizing data minimization over practical IoT machine-learning applications.

Notation: First, we define the following notation and terms.

- **Features**. Each feature f in a feature set F represents a distinct signal (modality) that can be used by the model. Features can be single-dimensional (scalar) or more complex.
- **Labeling**. A labeling is an assignment of each training example $x_i \in X$ in a dataset to a label $y_i \in Y$. Labels can be categorical, real-valued, or vector-valued.
- **Model**. A model is a predictor that can take as input $F' \subseteq F$ to predict some target label. That is, it is trained on a projection of $X[F']$ to predict some Y.

3.1 Re-identification Game

We formalize the data minimization problem as a two-player game consisting of a Provider and an Adversary. As the names imply, Provider is trying to solve some IoT task with machine learning and the Adversary is trying to re-identify users.

This game can be modeled as a hypothetical optimization that happens at training time. Both players have knowledge of the training dataset X which has two labelings Y_{task} and Y_{user}. Y_{task} describes a desired prediction target of the Provider. Y_{user} describes a user/entity/individual identifier the Provider is trying to hide. There is a preset parameter ℓ that both players know and an oracle that measures the accuracy of any model $acc(m)$. The game proceeds as follows:

1. The Provider selects a subset of features F' and presents a model m that predicts Y_{task} from the subset.
   - If $acc(m) < \ell$, the Provider automatically loses the game; a reward score of $-\infty$.
2. The Adversary observes F' and m, and presents a model m_{adv} that predicts Y_{user} from the same subset of features.
3. The Provider receives a reward score $-acc(m_{adv})$; the better the user-classification model, the worse the score.

In this game, the Provider is trying to maximize their reward score, which means finding a subset of features that does not degrade accuracy beyond ℓ (which can be calibrated against a model trained on the entire set F) while minimizing the accuracy of a potential adversary (i.e. identifiability). To be able to tractably identify a strategy for the Provider, we need to make a simplifying assumption:

Assumption 1 (Model Class Parity)**.** The Adversary chooses a model m' from the same pre-defined model family as the m given by the provider.

In other words, if the Provider can define a broad class of acceptable models as a part of their training procedure, e.g., random forests or linear models. m is the best model in that class that predicts Y_{task} that is yielded from parameter and hyperparameter optimization. Assumption 1 enforces that m' is found by the same procedure but against a different labeling. This assumption enforces a few desirable properties: (1) the Adversary is not substantially more capable than the Provider at prediction, (2) there is a bounded world of techniques that the Adversary can use, and (3) the Adversary is implicitly assuming that the Provider is a subject matter expert that has picked the best model for this feature space.

Of course, the above game is NP-Hard to solve, even in the simplified setting where each feature f has an identifiability score (cost) c_f and a utility score (value) v_f. Assuming that the utility and identifiability of each feature f are linearly additive, the optimal solution reduces to a 0-1 Knapsack problem:

$$\min \sum_{f \in F} c_f \qquad \text{s.t.} \qquad \sum_{f \in F} v_f \geq V$$

where V is a total utility threshold that controls the number of features selected. In practice, decomposing a feature space into identifiability and utility scores is very challenging. Complex ML problems often have multi-feature interactions, where the value of one feature is conditioned on the presence or absence of another.

3.2 Provider Strategy as an API

The rest of this paper shows how we turn the re-identification game into a feature optimization API that providers can use to achieve some data minimization. We assume that the service provider makes a best effort to protect user privacy; however, certain features may still act as side channels, leaking information that can be used to infer user identity. Thus, the optimal provider strategy from the game above gives a provider guidance on how to navigate this trade-off. The API is designed as follows:

```
def feature_minimize(features, utility_labels,
    user_labels, threshold) -> minimized_features
```

The user-defined threshold l (which is a calibrated version of ℓ above) is a number between 0 and 1, representing how much accuracy loss the user is willing to tolerate while optimizing for lower identifiability. Setting the threshold to 1 means that the user absolutely wants the lowest identifiability, even if the accuracy reaches 0%; a threshold of 0 means that no accuracy loss is acceptable, and the algorithm can only search for feature subsets that maintain the same accuracy as the original dataset. For instance, a threshold of 0.1 implies that the user is comfortable with 10% accuracy loss compared to its original level. The algorithm will then search for feature subsets as long as the accuracy remains at least 90% of the level before data minimization.

Definition 2 (Relative Effectiveness)**.** We define the relative effectiveness r_i as follows:

$$r_i = \log(\max(\frac{\text{Identifiability}_0 - \text{Identifiability}_i}{\text{Accuracy}_0 - \text{Accuracy}_i}, \epsilon))$$

where Identifiability$_i$ and Accuracy$_i$ represent the identifiability and accuracy at threshold i while Identifiability$_0$ and Accuracy$_0$ correspond to their values at threshold 0. This threshold corresponds to the parameter l defined in Definition 1. We include a small constant $\epsilon > 0$ to clip the ratio. This metric quantifies how sensitive identifiability is to changes in accuracy. A higher r_i indicates that identifiability can be significantly reduced with minimal accuracy loss, making the trade-off more favorable.

4 PRACTICAL SOLUTIONS TO THE RE-IDENTIFICATION GAME

Behind our API lies a practical algorithm implementing the Provider strategy. We begin with an optimal but computationally inefficient approach (§4.1), followed by a more efficient but suboptimal alternative (§4.2), and finally a hybrid solution that balances computational feasibility and optimality by combining the strengths of both (§4.3).

4.1 Simple Exhaustive Search

A straightforward strategy for the Provider is to perform an exhaustive search on all possible subsets of the feature set F. This involves training a primary model m and a user-classification model m_{adv} on each of the $2^{|F|}$ subsets. Then, we log all the accuracies for the primary model and the identifiabilities for the user-classification model. All these feature subsets are ranked based on the following priorities:

1. Identify the highest achievable primary model accuracy among all feature subsets and gradually relax it according to the user-defined threshold l.
2. Filter out feature subsets that do not meet the specified accuracy threshold.
3. Sort the remaining feature subsets in ascending order of user-classification model identifiability, prioritizing those with the lowest identifiability. The top result is our desired feature set with the best utility-identifiability tradeoff.

Although this approach guarantees optimality according to Definition 1, it is computationally infeasible for large $|F|$. Even though the process is perfectly parallelizable, exhaustive search on more than 15 features can take hours on a commodity server, making it impractical for high-dimensional datasets.

4.2 Greedy Selection

A more efficient but less optimal alternative to exhaustive search is to assume that each feature's contribution is linearly additive. Under this assumption, we can decompose accuracy and identifiability into per-feature values, allowing us to approximate feature importance without evaluating all possible subsets. The key idea is to develop a scoring mechanism that quantifies each feature's impact on both:

- The primary model's accuracy (utility score, or value).
- The user-classification model's accuracy (identifiability score, or cost).

As described in 2.2.3, several techniques can be used to estimate these scores. For example, we can apply the mutual information-based utility score and the entropy-based privacy score from [35] as our utility score and identifiability score here, respectively. Similarly, we can take the Gini impurity of both models and apply the feature importance values to each feature as utility and identifiability scores. As an alternative, we can also calculate the SHAP value for both models and represent them as scores as follows:

Let $\boldsymbol{S_{adv}}$ be the SHAP values for the user-classification model along the feature axis, and $\boldsymbol{S}$ be the SHAP values for the primary model along the feature axis.

- The utility score $\boldsymbol{v} = \text{mean}(\max(|\boldsymbol{S}|))$.
- The identifiability score $\boldsymbol{c} = \text{mean}(\max(|\boldsymbol{S_{adv}}|))$.

where the mean is over all data points and the max is taken over all output classes to capture the strongest feature contributions.

Once each feature is assigned a utility score (v) and an identifiability score (c), we can apply a greedy algorithm to solve the knapsack problem (as described in §3.1).

Greedy by utility. One approach to solving the problem is to prioritize utility by selecting features with the highest predictive power. We sort all features in descending order by their utility score v_f, and then iteratively select the top features until the utility constraint is met. A larger V allows more features to be retained, maintaining higher accuracy at the cost of higher identifiability.

Greedy by identifiability. Another approach is to prioritize privacy by minimizing identifiability. We sort all features in ascending order by identifiability score c_f, and then iteratively choose the least identifiable features until the utility constraint is met. A larger V allows more features to be selected, increasing both utility and identifiability.

Greedy by cost-to-value ratio. A more balanced approach is to consider the cost-to-value ratio, c_f/v_f, for each feature. Features are sorted in ascending order by this ratio, and the lowest-ratio features are selected iteratively until the utility constraint is met. This method strikes a balance between utility and privacy, prioritizing features that provide high predictive power with minimal identifiability risk. As before, V remains a tunable parameter that controls the number of features selected.

4.3 Hybrid Solution

Despite an efficient approach, the underlying assumption behind our greedy algorithm does not always hold. There are cases where features interact with each other and can be correlated. Selecting features independently based on their individual contributions can lead to suboptimal results. To balance efficiency and accuracy, we propose a two-stage solution:

1. Greedy Preselection (§4.2): We first apply a greedy algorithm to reduce the feature set to a tractable size. The user can specify a planned job duration for the exhaustive search, and the system automatically determines how many features to retain based on runtime estimates.
2. Exhaustive Search (§4.1): Within this reduced feature set, we then enumerate all possible combinations of the remaining features to identify the optimal subset.

This hybrid approach leverages the speed of greedy selection to narrow down candidates while ensuring optimality within the reduced feature space. §5.4.1 evaluates how our hybrid solution is better than greedy itself.

5 EVALUATION

5.1 Tasks and Datasets

We selected 7 datasets to evaluate the approaches we introduced in earlier sections. In all these tasks, there are a primary model and a user-classification (threat) model. The primary model makes predictions on the actual task the user cares about, whereas the threat model aims to infer the individual behind each request. We use random forest models by default for both primary and threat models. Table 1 summarizes the datasets we used in experiments, and is sorted by the number of features. The number of classes refers to the possible outcomes of the primary model and the number of users means the possible outcomes of the threat model. Out of the number of features shown in Table 1, our algorithm selects a subset of them to reach a trade-off between the primary model accuracy and the threat model identifiability.

Table 1: Datasets used in experiments.

Dataset	# classes	# users	# features
Aposemat IoT-23 [9]	2	31	9
IoT Sentinel [23]	31	333	22
Device Identification (NetML [43]) [19]	5	9	36
CIC-IDS [33]	2	271	74
Opportunity [29]	5	4	135
Device Identification (nPrint [12]) [19]	5	9	2667
Service Recognition (nPrint [12]) [2, 20, 22]	11	435	3285

Table 2: Comparisons between state-of-the-art and two-stage data minimization results.

Dataset	SOTA		Feat. Min.		Rel. Eff.
	Acc.	Ident.	Acc.	Ident.	
Device Ident. (NetML)	95.62%	69.52%	94.78%	52.82%	2.990
IoT Sentinel	70.60%	62.03%	69.96%	50.86%	2.860
CIC-IDS	97.98%	57.68%	97.02%	45.85%	2.511
Aposemat IoT-23	99.91%	77.98%	99.05%	69.81%	2.251
Opportunity	87.77%	99.87%	81.68%	66.46%	1.702
Device Ident. (nPrint)	98.12%	86.43%	92.48%	56.99%	1.652
Service Recogn. (nPrint)	95.52%	91.63%	84.95%	77.84%	0.266

Network intrusion detection [9, 33]. CIC-IDS 2017 and Aposemat IoT-23 are two network traffic datasets for intrusion detection analysis. The features are network traffic statistics such as duration, number of packets, number of bytes, length of packets, etc. The labels are binary (either benign or attack) but there are many possible IP addresses. We group them into IP subnets and use them as a proxy of users. Our goal is to select features that are predictive for the intrusion detection task with the least IP leakage.

Automated device-type identification for IoT [23]. IoT Sentinel is a networking dataset for IoT device-type identification. Its features are primarily derived from network headers and protocol metadata across all OSI layers, including protocol types, IP options, port usage, and packet sizes. Each sample is labeled with a specific device type, such as Hue Switch or TP-Link Plug. Compared to CIC-IDS 2017, IoT Sentinel includes fewer features but encompasses a broader range of device classes. As a result, the primary model achieves lower accuracy than models trained on CIC-IDS 2017.

Sensor-based human activity recognition [29]. Opportunity is a sensor-based dataset for human activity recognition. We use locomotion prediction (sit, walk, stand, lie) as the primary model and user prediction as the threat model.

Device identification [19]. We leverage the open-source dataset from AMIR [19] covering 5 in-home IoT devices. The label schema is similar to that of IoT Sentinel, but the dataset provides two distinct feature representations: a dense format from NetML [43] and a sparse format from nPrint [12]. A detailed ablation study comparing these two representations is presented in §5.4.2.

Table 3: Top 3 methods for each dataset.

Dataset	🥇	🥈	🥉
IoT Sentinel	(8) SHAP by CTV	(3) Privacy	(4) Utility
CIC-IDS 2017	(1) Hashing	(4) Utility	(8) SHAP by CTV
Opportunity	(4) Utility	(5) Tradeoff	(8) SHAP by CTV
Device Id. (NetML)	(4) Utility	(6) SHAP by utility	(5) Tradeoff
Device Id. (nPrint)	(4) Utility	(3) Privacy	(5) Tradeoff
Serv. Recog. (nPrint)	(4) Utility	(5) Tradeoff	(3) Privacy

Table 4: IoT-23 (9 feats).

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	99.91%	77.98%	N/A	5
0.01	**99.05%**	**69.81%**	**2.251**	**1**
0.03	99.05%	69.81%	2.251	1
0.1	95.15%	60.60%	1.295	1
0.3	95.15%	60.60%	1.295	1
1.0	95.15%	60.60%	1.295	1

Table 5: IoT Sentinel (13 feats).

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	70.66%	62.02%	N/A	10
0.01	**69.96%**	**50.86%**	**2.769**	**8**
0.03	69.03%	49.80%	2.014	4
0.1	63.61%	46.47%	0.791	4
0.3	49.93%	26.61%	0.535	2
1.0	21.34%	17.04%	-0.092	1

Table 6: CIC-IDS (15 feats).

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	97.85%	50.94%	N/A	5
0.01	**97.02%**	**45.85%**	**1.814**	**2**
0.03	95.14%	43.51%	1.009	2
0.1	91.10%	38.01%	0.650	1
0.3	70.70%	37.18%	-0.680	1
1.0	70.70%	37.18%	-0.680	1

Table 7: Opportunity (10 feats).

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	82.43%	69.21%	N/A	10
0.01	81.68%	66.46%	1.299	8
0.03	**80.10%**	**60.11%**	**1.362**	**7**
0.1	75.40%	50.95%	0.955	5
0.3	58.39%	30.51%	0.476	3
1.0	42.76%	26.14%	0.082	2

Table 8: Opportunity (7 feats).

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	78.91%	61.50%	N/A	7
0.01	78.31%	60.94%	-0.069	6
0.03	76.71%	57.04%	0.707	6
0.1	74.16%	50.28%	0.860	4
0.3	58.53%	31.89%	0.374	3
1.0	42.74%	26.17%	-0.023	1

Table 9: Opportunity (5 feats).

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	77.05%	57.39%	N/A	5
0.01	77.05%	57.39%	N/A	5
0.03	77.05%	57.39%	N/A	5
0.1	69.91%	49.40%	0.112	3
0.3	57.14%	33.51%	0.182	2
1.0	42.74%	26.17%	-0.094	1

Service recognition [2, 20, 22]. We aggregate three datasets of 23,487 flows in total as our service recognition task. The labels correspond to network application types, including video conferencing (e.g., Zoom, Google Meet), video streaming (e.g., YouTube, Twitch), and social media (e.g., Instagram, Facebook). In our primary model, those combined flows are categorized into 11 classes of applications. This dataset only comes in nPrint [12] sparse representations.

5.2 Ranking Feature Selection Methods

For greedy selection, we employ the following feature selection methods on aforementioned datasets: (1) feature hashing, (2) PCA, (3) entropy-based privacy score, (4) mutual information-based utility score, (5) privacy-utility tradeoff score, (6) SHAP-based greedy by utility, (7) SHAP-based greedy by identifiability, (8) SHAP-based greedy by cost-to-value ratio, (9) Gini impurity-based greedy by utility, (10) Gini impurity-based greedy by identifiability, (11) Gini impurity-based greedy by cost-to-value ratio, and (12) differential privacy. Detailed description of each method can be found in §2.2.

We rank these methods based on the highest relative effectiveness, and Table 3 shows the top 3 methods for each dataset. Although there is no clear winner, the best performing methods are (4) utility score and (8) SHAP-based greedy by CTV across different datasets. Traditional feature selection methods, such as PCA, do not perform well. For more detailed comparisons between feature selection methods, a case study can be found in §5.5.

Taking exhaustive search into account as well, Table 2 shows a summary of our two-stage data minimization results, and is sorted by relative effectiveness. The base accuracy and identifiability for relative effectiveness come from the full dataset without data minimization. 10-15 features are selected from the greedy stage by SHAP or utility score, and 0.01 is used as the threshold in the exhaustive search stage.

5.3 Effectiveness of Exhaustive Search

5.3.1 Exhaustive search works well for a dataset with a small number of features. Given a dataset with tractable dimensionality, our exhaustive search algorithm described in §4.1 provides the optimal accuracy-identifiability tradeoff by enumerating all possible feature subsets. On the Aposemat IoT-23 dataset (Table 4), we find that retaining only the "orig_pkts" feature on cleaned data (without NaNs) achieves an 8% reduction in identifiability with less than 1% accuracy loss. If we relax the threshold, another feature is chosen and the identifiability can further be reduced to 60.60% while maintaining 95% primary model accuracy.

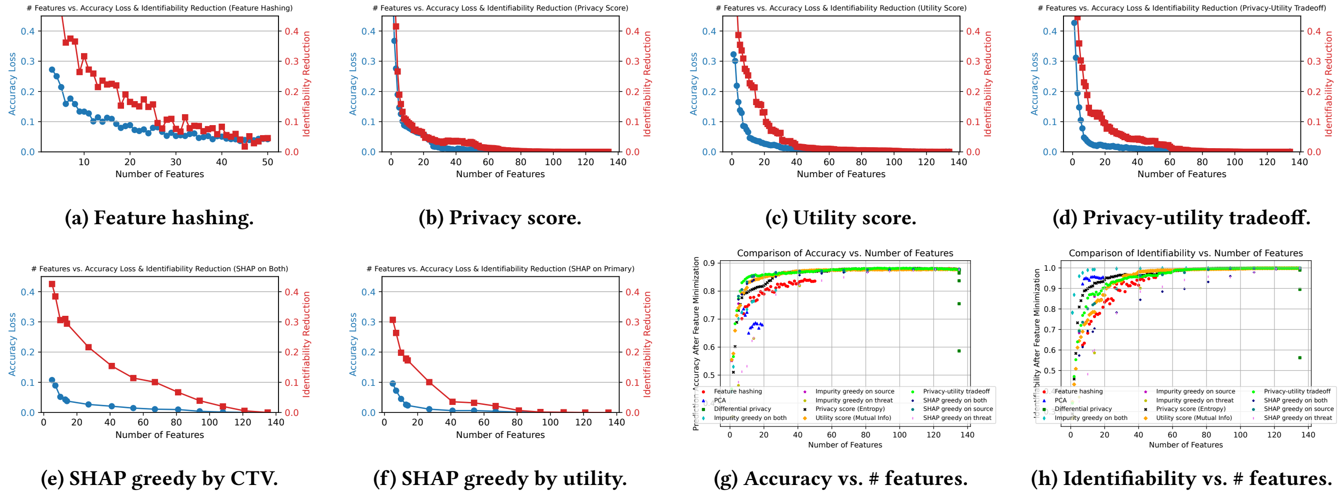
5.3.2 For larger datasets with dense features, our two-stage hybrid solution achieves good tradeoffs. When the dataset contains a larger number of features, exhaustive search becomes computationally infeasible due to its exponential complexity. In these cases, we resort to the greedy algorithm (§4.2) to reduce the feature space, followed by exhaustive search on the remaining features. Table 5, 6, 7 illustrate how our two-stage approach achieves a favorable tradeoff between model accuracy and identifiability across various telemetry-based datasets. Here, the base accuracy and identifiability for relative effectiveness come from a 0-threshold exhaustive search after the greedy stage. Empirically, setting the threshold to 0.01 appears to be a reasonable choice across multiple datasets.

5.4 Ablation Study

The purpose of our greedy algorithm (§4.2) is to reduce the number of features to at most 10–15 while retaining as much information as possible. Ideally, the reduced feature subset should be informative enough to allow a subsequent exhaustive search to find the optimal tradeoff between accuracy and identifiability.

5.4.1 Reducing the number of features in exhaustive search will lead to worse tradeoffs. Since exhaustive search is so computationally expensive, why not rely solely on greedy selection? As shown in Fig. 1c, a substantial accuracy loss is often observed when the number of features retained by greedy preselection is small. This sudden drop is why we adopt exhaustive search as a second stage, allowing fine-grained optimization of the selected feature subset.

To assess the impact of greedy preselection on the exhaustive search stage, we vary the number of features retained by the greedy algorithm and compare the corresponding performance in our two-stage solution. By examining Table 7, Table 8, and Table 9, we observe that the best tradeoff between accuracy and identifiability is achieved when 10 features are retained by the greedy preselection stage, compared to 5 or 7 features. This is because retaining more features allows for greater flexibility in the subsequent exhaustive search, enabling it to find an optimal subset that balances accuracy



(a) Feature hashing. **(b) Privacy score.** **(c) Utility score.** **(d) Privacy-utility tradeoff.**

(e) SHAP greedy by CTV. **(f) SHAP greedy by utility.** **(g) Accuracy vs. # features.** **(h) Identifiability vs. # features.**

Figure 1: Case study: Opportunity dataset (greedy selection).

Table 10: Dev. Ident. (NetML)

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	95.62%	56.37%	N/A	4
0.01	**94.78%**	**52.82%**	**1.441**	**3**
0.03	93.53%	49.27%	1.223	4
0.1	86.22%	42.17%	0.413	3
0.3	73.90%	34.03%	0.028	1
1.0	73.90%	34.03%	0.028	1

Table 11: Dev. Ident. (nPrint)

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	93.32%	59.71%	N/A	10
0.01	**92.48%**	**56.99%**	**1.175**	**7**
0.03	90.61%	54.49%	0.656	8
0.1	84.34%	47.60%	0.299	7
0.3	66.81%	36.53%	-0.134	2
1.0	46.76%	31.52%	-0.502	1

Table 12: Serv. Recog. (nPrint)

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	85.65%	78.83%	N/A	10
0.01	**84.95%**	**77.84%**	**0.347**	**8**
0.03	83.27%	75.94%	0.194	8
0.1	77.16%	71.48%	-0.144	5
0.3	60.33%	57.39%	-0.166	3
1.0	37.60%	40.65%	-0.230	1

Table 13: Opp. (10 feats, 50%)

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	81.24%	65.31%	N/A	10
0.01	80.56%	62.54%	1.405	9
0.03	**79.79%**	**59.13%**	**1.450**	**8**
0.1	73.15%	48.39%	0.738	4
0.3	58.10%	30.03%	0.422	3
1.0	42.23%	25.77%	0.013	1

Table 14: Opp. (10 feats, 10%)

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	76.74%	50.95%	N/A	8
0.01	**76.09%**	**48.73%**	**1.228**	**8**
0.03	75.10%	45.53%	1.195	7
0.1	71.32%	40.34%	0.672	6
0.3	56.13%	28.61%	0.081	3
1.0	37.39%	25.60%	-0.440	1

Table 15: Opp. (MLP, 10 feats)

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	77.71%	53.34%	N/A	7
0.01	**77.08%**	**49.75%**	**1.740**	**8**
0.03	75.44%	45.49%	1.241	6
0.1	70.40%	36.27%	0.848	4
0.3	60.50%	28.28%	0.376	3
1.0	46.90%	26.49%	-0.138	1

Table 16: Opp. (KNN, 10 feats)

Thr.	Acc.	Ident.	Rel. Eff.	# Feat.
0.0	78.32%	63.62%	N/A	7
0.01	**77.57%**	**59.21%**	**1.772**	**9**
0.03	76.43%	54.66%	1.556	7
0.1	72.51%	47.23%	1.037	5
0.3	56.19%	27.21%	0.498	2
1.0	40.45%	25.48%	0.007	1

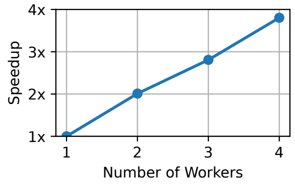


Figure 2: Scalability.

and privacy more effectively. However, retaining more features means that more computation is required for the exhaustive search.

5.4.2 *Dense feature representations lead to better tradeoffs than sparse representations.*

For the device identification dataset, we evaluated two different featurization of the same data: NetML [43] and nPrint [12]. NetML is a featurized representation with 36 dense features, where each feature captures higher-level statistical properties of network traffic. nPrint, in contrast, represents the raw bit-level packet data with 2667 sparse features, where each feature can take values 1, 0, or -1 (with -1 indicating non-existent headers in a given packet). Unlike dense feature representations, sparse feature sets distribute information across a large number of low-information features, meaning that each individual feature carries limited predictive value. We find that our two-stage method performs well on the NetML representation but is less effective on nPrint's sparse representation (Table 10, Table 11).

Since the complexity of exhaustive search grows exponentially with the number of features retained by greedy selection, sparse feature sets present a challenge: less information is contained in the same number of features compared to dense representations. As a result, even though exhaustive search attempts to find the optimal subset, it is less effective in sparse settings because the optimization space is inherently limited. This limitation is more evident in the service recognition dataset (Table 12), where relative effectiveness remains low despite reductions in both accuracy and identifiability. This is expected, as *sparse features tend to contribute to both accuracy and identifiability simultaneously. They are both predictive and revealing, making the tradeoff less favorable.* Thus, our two-stage method performs better on datasets with dense features, where more meaningful tradeoffs between identifiability and accuracy can be achieved.

While in theory, a better utility-identifiability tradeoff could be found using sparse feature representations like nPrint given unlimited computational resources, the sheer number of possible feature subsets makes exhaustive search impractical. The potential improvement is unlikely to justify the additional computational cost, making dense feature representations a more practical choice for data minimization in real-world applications.

5.4.3 *Our method scales with both the number of rows and workers.*

Table 13 and Table 14 follow the same two-stage experimental setup as Table 7, with the only difference being that 50% and 10% of the data, respectively, are used to train both the primary and threat models. While both accuracy and identifiability decrease slightly, the drop is not substantial, and our method still achieves a reasonable level of relative effectiveness.

As described in §4, the exhaustive search is the primary computational bottleneck, while the greedy solution is relatively fast to run. Fig. 2 illustrates the scalability of exhaustive search on the Opportunity dataset. Since the search space can be easily partitioned by distributing feature subsets evenly across workers, our approach achieves near-linear speedup through parallelization.

5.4.4 Our method is transferrable across different models. In all aforementioned experiments, we have used random forest as both the primary and threat models for consistency. To assess the robustness of our findings across different model architectures, Table 15 and Table 16 present results from applying our two-stage data minimization approach using multi-layer perceptron (MLP) and k-nearest neighbors (KNN), respectively, on the same Opportunity dataset. The results are largely consistent with those obtained using random forest, indicating that our conclusions generalize well across different model choices.

5.5 Case Study: Opportunity Dataset

Fig. 1 presents a comprehensive analysis of various greedy selection methods applied to the Opportunity human activity recognition dataset. The key observations are summarized below.

5.5.1 Identifiability reduction exceeds accuracy loss. In Fig. 1a–1f, we compare identifiability reduction (right y-axis) against accuracy loss (left y-axis) across various feature selection methods. The accuracy loss (blue line with rounded dots) is generally lower than the identifiability reduction (red line with squared dots) for a given number of selected features. This indicates that our feature selection methods effectively reduce identifiability while keeping accuracy degradation minimal, validating our approach for privacy-aware data minimization.

5.5.2 Spike in accuracy loss when too few features are retained. When the number of features becomes too small, both accuracy loss and identifiability reduction increase sharply, as shown in Fig. 1a–1f, 1g, 1h. This is problematic because accuracy loss at this level is typically unacceptable, reinforcing the need to perform exhaustive search for finer-grained optimization with a reasonable number of features. Across all feature selection methods, performance becomes less stable when the number of features is reduced too aggressively, highlighting the importance of our two-stage hybrid solution.

5.5.3 Limitations of relative effectiveness. While relative effectiveness is a useful metric for evaluating identifiability reduction per unit of accuracy loss, it has two key limitations: (1) The metric can be amplified when accuracy loss is negligible, making the tradeoff appear more favorable than it actually is. (2) Even when relative effectiveness appears high, both accuracy and identifiability may have declined significantly, potentially leading to unacceptable accuracy for practical use.

In Fig. 1c, relative effectiveness is the highest when 103 features are selected. However, both accuracy drop and identifiability reduction at that point are minimal, which inflates the metric. Conversely, although the left tail (i.e., when few features are retained) may still yield seemingly reasonable relative effectiveness, the accompanying accuracy degradation is often too severe to be acceptable.

5.6 Recommendations for Best Practices

Based on our experiments, we recommend a structured approach for effective data minimization:

Choose the right feature representations: prioritize dense representations over sparse alternatives. When initiating data minimization, dense representations (e.g., NetML) are preferred over sparse representations (e.g., nPrint). Dense features inherently encode more information per feature, enabling computationally efficient optimization while effectively reducing identifiability. This contrasts with sparse representations, which often require retaining larger feature sets for utility, conflicting with minimization goals.

Pre-select features: use mutual information and SHAP-based methods. Mutual information-based feature utility scores and SHAP-based greedy selection using cost-to-value ratio (CTV) are two effective methods for preliminary data minimization. Our findings also indicate that traditional feature selection methods (e.g., PCA, feature hashing) *should be avoided for privacy-aware feature selection.* Depending on the dataset size and available computational resources, retaining *10–15 features* is recommended to achieve better overall relative effectiveness in the next stage.

Minimize identifiability by exhaustive search: a 0.01 threshold offers a practical tradeoff. In our experiments, setting the accuracy loss tolerance parameter $l = 0.01$ appears to achieve a favorable balance between predictive performance and user identifiability in most cases. This threshold allows for meaningful reductions in identifiability while keeping the impact on predictive accuracy minimal. While other applications may tolerate larger accuracy losses, we find that a small accuracy drop (e.g., 1%) is generally acceptable and aligns with the overarching goal of maintaining functional model performance.

6 DISCUSSION AND FUTURE WORK

Table 3 confirms that neither feature hashing nor PCA improves performance when applied to sparse features (e.g., nPrint [12]). This is because each coordinate in the nPrint vector merely records the presence of a single packet bit; projecting such sparse binary vectors does not introduce new, semantically rich signals. In this context, "densification" functions more as a compression technique than as a method of feature selection. A straightforward remedy is to re-encode the same traffic with a denser feature set, such as the flow-level statistical vectors provided by NetML [43], where each attribute aggregates many bytes and thus carries richer variation.

Beyond manual feature engineering, foundation models for packet data now offer a direct path to densification: netFound [11] learns multimodal embeddings from large unlabeled traces, and NetLLM [41] shows that such embeddings can be fine-tuned quickly for downstream intrusion-detection and classification tasks. Mapping nPrint outputs into these learned embedding spaces would supply the dense features that our results suggest are necessary.

REFERENCES

[1] Mina Alishahi, Vahideh Moghtadaiee, and Hojjat Navidan. 2022. Add noise to remove noise: Local differential privacy for feature selection. *Computers & Security* 123 (2022), 102934. https://doi.org/10.1016/j.cose.2022.102934

[2] Francesco Bronzino, Paul Schmitt, Sara Ayoubi, Guilherme Martins, Renata Teixeira, and Nick Feamster. 2019. Inferring Streaming Video Quality from Encrypted Traffic: Practical Models and Deployment Experience. *Proc. ACM Meas. Anal. Comput. Syst.* 3, 3, Article 56 (Dec. 2019), 25 pages. https://doi.org/10.1145/3366704

[3] California State Legislature. 2018. California Consumer Privacy Act of 2018. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375 Assembly Bill No. 375.

[4] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, Shai Halevi and Tal Rabin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 265–284.

[5] David Eklund, Alfonso Iacovazzi, Han Wang, Apostolos Pyrgelis, and Shahid Raza. 2024. BMI: Bounded Mutual Information for Efficient Privacy-Preserving Feature Selection. In *Computer Security – ESORICS 2024*, Joaquin Garcia-Alfaro, Rafał Kozik, Michał Choraś, and Sokratis Katsikas (Eds.). Springer Nature Switzerland, Cham, 353–373.

[6] Abdussalam Elhanashi, Pierpaolo Dini, Sergio Saponara, and Qinghe Zheng. 2023. Integration of Deep Learning into the IoT: A Survey of Techniques and Challenges for Real-World Applications. *Electronics* 12, 24 (2023). https://doi.org/10.3390/electronics12244925

[7] European Parliament and Council of the European Union. [n.d.]. Regulation (EU) 2016/679 of the European Parliament and of the Council. https://data.europa.eu/eli/reg/2016/679/oj

[8] Prakhar Ganesh, Cuong Tran, Reza Shokri, and Ferdinando Fioretto. 2024. The Data Minimization Principle in Machine Learning. In *Proceedings of the 41st International Conference on Machine Learning. 2nd Workshop on Generative AI and Law (GenLaw '24).* (Messe Wien Exhibition Congress Center, Vienna, Austria). 21–27. https://blog.genlaw.org/pdfs/genlaw_icml2024/33.pdf

[9] Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga. 2021. *IoT-23: A labeled dataset with malicious and benign IoT network traffic.* https://doi.org/10.5281/zenodo.4743746

[10] Corrado Gini. 1912. *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche.[Fasc. I.].* Tipogr. di P. Cuppini.

[11] Satyandra Guthula, Roman Beltiukov, Navya Battula, Wenbo Guo, and Arpit Gupta. 2023. netFound: Foundation model for network security. *arXiv preprint arXiv:2310.17025* (2023).

[12] Jordan Holland, Paul Schmitt, Nick Feamster, and Prateek Mittal. 2021. New Directions in Automated Traffic Analysis *(CCS '21).* Association for Computing Machinery, New York, NY, USA, 3366–3383. https://doi.org/10.1145/3460120.3484758

[13] Sowmya Jagadeesan, C.N. Ravi, M. Sujatha, S Sree Southry, J Sundararajan, and Ch. Venkata Krishna Reddy. 2023. Machine Learning and IoT based Performance Improvement of Energy Efficiency in Smart Buildings. In *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS).* 375–380. https://doi.org/10.1109/ICSCDS56580.2023.10104874

[14] Xi Jiang, Shinan Liu, Saloua Naama, Francesco Bronzino, Paul Schmitt, and Nick Feamster. 2023. Ac-dc: Adaptive ensemble classification for network traffic identification. *arXiv preprint arXiv:2302.11718* (2023).

[15] Ye-Seul Kil, Yeon-Ji Lee, So-Eun Jeon, Ye-Sol Oh, and Il-Gu Lee. 2024. Optimization of Privacy-Utility Trade-Off for Efficient Feature Selection of Secure Internet of Things. *IEEE Access* 12 (2024), 142582–142591. https://doi.org/10.1109/ACCESS.2024.3467049

[16] J Kreer. 1957. A question of terminology. *IRE Transactions on Information Theory* 3, 3 (1957), 208–208.

[17] Wei Li, Yuanbo Chai, Fazlullah Khan, Syed Rooh Ullah Jan, Sahil Verma, Varun G. Menon, Kavita, and Xingwang Li. [n.d.]. A Comprehensive Survey on Machine Learning-Based Big Data Analytics for IoT-Enabled Smart Healthcare System. 26, 1 ([n. d.]), 234–252. https://doi.org/10.1007/s11036-020-01700-6

[18] Shinan Liu, Francesco Bronzino, Paul Schmitt, Arjun Nitin Bhagoji, Nick Feamster, Hector Garcia Crespo, Timothy Coyle, and Brian Ward. 2023. Leaf: Navigating concept drift in cellular networks. *Proceedings of the ACM on Networking* 1, CoNEXT2 (2023), 1–24.

[19] Shinan Liu, Tarun Mangla, Ted Shaowang, Jinjin Zhao, John Paparrizos, Sanjay Krishnan, and Nick Feamster. 2023. AMIR: Active Multimodal Interaction Recognition from Video and Network Traffic in Connected Environments. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 1, Article 21 (mar 2023). https://doi.org/10.1145/3580818

[20] Shinan Liu, Ted Shaowang, Gerry Wan, Jeewon Chae, Jonatas Marques, Sanjay Krishnan, and Nick Feamster. 2024. ServeFlow: A Fast-Slow Model Architecture for Network Traffic Analysis. *arXiv preprint arXiv:2402.03694* (2024).

[21] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 4765–4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[22] Kyle MacMillan, Tarun Mangla, James Saxon, and Nick Feamster. 2021. Measuring the performance and network utilization of popular video conferencing applications. In *Proceedings of the 21st ACM Internet Measurement Conference* (Virtual Event) *(IMC '21).* Association for Computing Machinery, New York, NY, USA, 229–244. https://doi.org/10.1145/3487552.3487842

[23] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, Nadarajah Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. 2017. Iot sentinel: Automated device-type identification for security enforcement in iot. In *2017 IEEE 37th international conference on distributed computing systems (ICDCS).* IEEE, 2177–2184.

[24] John Moody. 1988. "Fast learning in multi-resolution hierarchies". In *Proceedings of the 2nd International Conference on Neural Information Processing Systems (NIPS'88).* MIT Press, Cambridge, MA, USA, 29–39.

[25] Shinji Ono, Jun Takata, Masaharu Kataoka, Tomohiro I, Kilho Shin, and Hiroshi Sakamoto. 2022. Privacy-Preserving Feature Selection with Fully Homomorphic Encryption. *Algorithms* 15, 7 (2022). https://doi.org/10.3390/a15070229

[26] Qi Pang, Lun Wang, Shuai Wang, Wenting Zheng, and Dawn Song. 2023. Secure Federated Correlation Test and Entropy Estimation. In *Proceedings of the 40th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (Eds.), Vol. 202. PMLR, 26990–27010. https://proceedings.mlr.press/v202/pang23a.html

[27] Karl Pearson. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2, 11 (1901), 559–572. https://doi.org/10.1080/14786440109462720 arXiv:https://doi.org/10.1080/14786440109462720

[28] Rudi Poepsel-Lemaitre, Kaustubh Beedkar, and Volker Markl. 2024. Disclosure-Compliant Query Answering. *Proc. ACM Manag. Data* 2, 6, Article 233 (Dec. 2024), 28 pages. https://doi.org/10.1145/3698808

[29] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and José del R. Millàn. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS).* 233–240. https://doi.org/10.1109/INSS.2010.5573462

[30] Ted Shaowang, Nilesh Jain, Dennis D Matthews, and Sanjay Krishnan. 2021. Declarative data serving: the future of machine learning inference on the edge. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2555–2562.

[31] Ted Shaowang and Sanjay Krishnan. 2023. EdgeServe: A Streaming System for Decentralized Model Serving. *arXiv preprint arXiv:2303.08028* (2023).

[32] Ted Shaowang, Xi Liang, and Sanjay Krishnan. 2022. Sensor Fusion on the Edge: Initial Experiments in the EdgeServe System. In *Proceedings of The International Workshop on Big Data in Emergent Distributed Environments* (Philadelphia, Pennsylvania) *(BiDEDE '22).* Association for Computing Machinery, New York, NY, USA, Article 8, 7 pages. https://doi.org/10.1145/3530050.3532924

[33] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. 2018. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy, ICISSP 2018, Funchal, Madeira - Portugal, January 22-24, 2018*, Paolo Mori, Steven Furnell, and Olivier Camp (Eds.). SciTePress, 108–116. https://doi.org/10.5220/0006639801080116

[34] Tanusree Sharma, Lin Kyi, Yang Wang, and Asia J. Biega. 2024. "I'm not convinced that they don't collect more than is necessary": User-Controlled Data Minimization Design in Search Engines. In *33rd USENIX Security Symposium (USENIX Security 24).* USENIX Association, Philadelphia, PA, 2797–2812. https://www.usenix.org/conference/usenixsecurity24/presentation/sharma

[35] Mina Sheikhalishahi and Fabio Martinelli. 2017. Privacy-Utility Feature Selection as a Privacy Mechanism in Collaborative Data Classification. In *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE).* 244–249. https://doi.org/10.1109/WETICE.2017.15

[36] Cuong Tran and Ferdinando Fioretto. 2023. Data Minimization at Inference Time. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 72248–72269. https://proceedings.neurips.cc/paper_files/paper/2023/file/e48880ea81caa7836e6a0694049093ae-Paper-Conference.pdf

[37] Cuong Tran and Ferdinando Fioretto. 2023. Data Minimization at Inference Time. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 72248–72269. https://proceedings.neurips.cc/paper_files/paper/2023/file/e48880ea81caa7836e6a0694049093ae-Paper-Conference.pdf

[38] Gerry Wan, Shinan Liu, Francesco Bronzino, Nick Feamster, and Zakir Durumeric. 2024. CATO: End-to-End Optimization of ML-Based Traffic Analysis Pipelines. *arXiv preprint arXiv:2402.06099* (2024).

[39] Chen Wang, Xiangdong Huang, Jialin Qiao, Tian Jiang, Lei Rui, Jinrui Zhang, Rong Kang, Julian Feinauer, Kevin A. McGrail, Peng Wang, Diaohan Luo, Jun Yuan, Jianmin Wang, and Jiaguang Sun. 2020. Apache IoTDB: time-series database

for internet of things. *Proc. VLDB Endow.* 13, 12 (Aug. 2020), 2901–2904. https://doi.org/10.14778/3415478.3415504

[40] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, Quebec, Canada) *(ICML '09)*. Association for Computing Machinery, New York, NY, USA, 1113–1120. https://doi.org/10.1145/1553374.1553516

[41] Duo Wu, Xianda Wang, Yaqi Qiao, Zhi Wang, Junchen Jiang, Shuguang Cui, and Fangxin Wang. 2024. Netllm: Adapting large language models for networking. In *Proceedings of the ACM SIGCOMM 2024 Conference*. 661–678.

[42] Yongji Wu, Matthew Lentz, Danyang Zhuo, and Yao Lu. 2022. Serving and Optimizing Machine Learning Workflows on Heterogeneous Infrastructures. *Proc. VLDB Endow.* 16, 3 (Nov. 2022), 406–419. https://doi.org/10.14778/3570690.3570692

[43] Kun Yang, Samory Kpotufe, and Nick Feamster. 2020. A Comparative Study of Network Traffic Representations for Novelty Detection. (2020). arXiv:2006.16993 [cs.NI]

[44] Weisi Yang, Shinan Liu, Feng Xiao, Nick Feamster, and Stephen Xia. 2025. Towards Scalable Defenses against Intimate Partner Infiltrations. *arXiv preprint arXiv:2502.03682* (2025).

[45] Hui Zou, Trevor Hastie, and Robert Tibshirani. 2006. Sparse principal component analysis. *Journal of computational and graphical statistics* 15, 2 (2006), 265–286.