Balancing Utility, Privacy, and Energy in Internet of Things Systems

by

Henrique Pötter

B.S. in Computer Science, Rio de Janeiro State University, 2012

M.S. in Computer Science, Rio de Janeiro State University, 2015

Submitted to the Graduate Faculty of

the Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2023

UNIVERSITY OF PITTSBURGH

DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Henrique Pötter

It was defended on

July 20th 2023

and approved by

Daniel Mossé, Advisor Department of Computer Science

Stephen Lee, Co-advisor Department of Computer Science

Adam Lee, Department of Computer Science

Daniel Cole, Swanson School of Engineering

Copyright © by Henrique Pötter
2023

Balancing Utility, Privacy, and Energy in Internet of Things Systems

Henrique Pötter, PhD

University of Pittsburgh, 2023

As the Internet of Things (IoT) enters consumer markets, smart devices with diverse sensing capabilities and always-on connectivity have become more accessible to the public. These devices bring automation and data-driven insights, but their widespread presence increases the risk of exposing private or confidential information. A common approach to protect privacy is using privacy-preserving solutions such as data obfuscation, but this approach has drawbacks. It might bolster privacy but also compromise data's utility. Also, it can demand additional energy, affecting the mobility of battery or energy-harvesting IoT device deployments.

This dissertation contributes to designing, implementing, and evaluating energy-efficient utility-aware privacy solutions to enable IoT systems to protect privacy and improve reliability. We study the IoT Utility, Privacy, and Energy (UPE) tradeoffs in three phases: to (1) define the requirements for privacy solutions to better balance the UPE tradeoffs; (2) understand the limitations of privacy solutions in the context of federated learning applications; and (3) preserve user privacy through the selective removal of only the sensitive contents of data.

In the first phase, we develop a new methodology to evaluate the UPE tradeoffs of privacy-preserving techniques by augmenting the conventional Utility-Privacy problem by adding energy consumption. This model is evaluated with two data modalities: image classification and audio applications. In phase two, we develop a methodology to assess the privacy guarantees of neural network inferences using differential privacy with federated learning for IoT. Lastly, in the third phase, we seek to minimize energy consumption by developing a solution to only target the most sensitive data segments. Here we create PrivSpeech, a framework that uses a lightweight neural network that only obfuscates the sensitive attributes while maintaining the utility with minimal energy consumption. We evaluate PrivSpeech with interchanging privacy and utility setups with models for gender

identification, emotion detection, and speaker verification.

Our research extends the current understanding of utility, privacy, and energy consumption in the IoT landscape, offering new methodologies and privacy-preserving solutions. We expect to contribute to IoT systems designers, assisting them in making informed decisions to ensure privacy in an efficient and utility-preserving manner to IoT applications.

Table of Contents

List of Tables

List of Figures

Preface

This dissertation stands as a testament to the work I have undertaken and the unwavering support, guidance, and camaraderie of those who have journeyed with me. To my esteemed advisers, I extend my deepest gratitude. Thank you for the countless hours reviewing and improving our work. Your profound knowledge and relentless guidance have shaped my personal growth. Dr. Daniel Mosse, thank you for all the time, support, and generosity in sharing your expertise over the years. Also, Dr. Stephen Lee, thank you for all your support, patience, and always availability.

Thank Dr. Adam J. Lee for the precious discussions on Privacy and Dr. Daniel Cole for teaching the importance of consistently delivering at least one thing every week. Also, thank you, both professors, for serving on my committee.

To my dear family, your unwavering support has been my bedrock. I want to thank my parents, Alberto Pötter and Elinei Pötter. You have stood by me, steadfast and nurturing, providing a stable foundation even when the road was fraught with challenges. To my beloved wife, Marilu Nunez, who shared the weight and daily effort necessary through all these years. To my now 6-month-old Daughter Emma Pötter, you are my hope for a better future. Thank you to all my family; you have celebrated my victories, however small. This achievement is as much yours as it is mine.

My friends, you've made this journey enjoyable, even in its most challenging moments. The camaraderie we have shared, the thought-provoking discussions, the late-night study, and the moments of light-hearted respite have all enriched my experience immeasurably.

Our understanding of the world and ourselves is an outcome of countless lifetimes of inquiry, exploration, and discovery. Each question asked, each theory proposed, each experiment conducted, each failure experienced, and each success celebrated by our ancestors becomes the platform where we stand, casting our gaze toward the uncharted frontiers of understanding. Each of you has contributed to this achievement uniquely, and I am profoundly grateful for that. In acknowledgment of our shared endeavor, I dedicate this dissertation to you.

1.0 Introduction

1.1 Motivation and Problem Statement

The transition of IoT into a commodity market has enabled cheap internet-connected devices with diverse sensing capabilities to expose geographically specific data about the physical world and the people around them [136, 42]. Alongside the benefits of automation and data insights from IoT applications, the increased use of sensors and data collection also increased the risk of collecting private or confidential data. Most users are unaware of how much data their devices and applications demand to carry out the functionalities that they expect. It has been shown that applications often collect much more data than strictly needed [50, 102, 104], typically processing this data in the cloud [122, 110]. Consequently, IoT service providers can have access to data about users that exceeds applications' real needs [122, 110]. If this data is leaked or repurposed, maliciously or by mistake, as it often happens [68, 55], the user's privacy is eventually breached.

In parallel with the proliferation of IoT, machine learning and data analytics tools are now more viable and accessible to researchers, businesses, and the public at large [14]. The availability of frameworks and libraries such as Tensorflow, OpenCV, and Pytorch, coupled with repurposed GPU cards from the video editing and gaming industry, have democratized machine learning applications, eliminating the need for costly specialized computing clusters or in-depth machine learning algorithm expertise [92, 79]. This broad adoption of artificial intelligence has made advanced data mining techniques more accessible, enabling the identification and use of patterns within seemingly arbitrary data for prediction systems in various applications. Thus, while machine learning is poised to exploit patterns in IoT data to uncover unknown and beneficial relationships, it also risks being misused by malicious entities to invade and acquire private and confidential information.

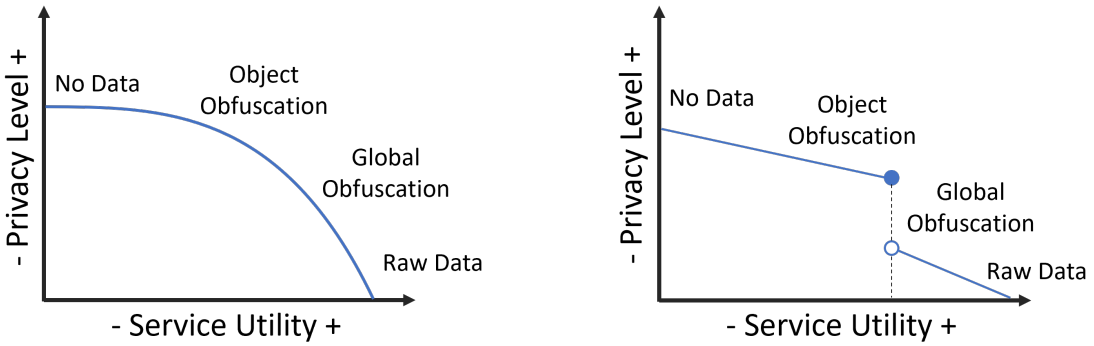
The human voice, for example, has become a critical component in voice user interfaces, enabling seamless engagement with intelligent assistants, state-of-the-art automotive systems, and a variety of other sophisticated devices [52, 62, 101]. This voice data carries

unique personal information, encapsulating insights into health status, emotional fluctuations, and more [130]. Consequently, any unauthorized distribution or leakage of this voice data poses a significant risk to the individual, emphasizing the need for protective measures.

Currently, there is a dependence on remote services in the emergent market of *app*ified IoT platforms, such as Samsung SmartThings [7] and Apple HomeKit [67]. These platforms allow developers to use data from smart home devices to create applications and rely on permission-based access control (PBAC) mechanisms to guarantee users' data protection. However, if raw sensor data is leaked, PBAC cannot protect against secondary data usage. For example, user location, behavioral patterns, or preference profiles can be inferred or directly extracted from different IoT devices [123, 113, 53]. The risks are even worse from IoT/smart devices that can be easily bought on popular platforms such as Amazon from unknown vendors/manufacturers; several of these devices have been shown to use protocols that share data without any encryption [40]. Hence, existing solutions are insufficient, making near-data processing (on-device or at the edge) necessary for protecting privacy [21].

Previous research has proposed different techniques to obfuscate users' information to protect IoT data privacy risks in the cloud [21, 124, 94, 20, 43, 111, 30]. However, reducing the amount of data exposure is not trivial. Sensor-embedded devices (e.g., cameras and voice assistants) can collect rich *unstructured data* (i.e., uncategorized data, such as videos and audio) data that can capture subtle patterns. These patterns can correlate with various other events and physical properties. For example, it has been shown that audio can be extracted from soundless video by observing how sound waves create vibrations on thin surfaces [39]. Removing such minute details from sensed data requires intelligent algorithms because blunt solutions, such as reducing the overall amount of data (e.g., decreasing the sensor sampling rate) or blurring an image, will often degrade the quality of expected services from the data (Utility). This problem is typically depicted as the Utility and Privacy (UP) tradeoff Pareto frontier [20].

Figure 1-(a) shows the typical asymmetric relationship between Utility and Privacy of data [141, 20, 104]. Intuitively, assuming that both the user's service (i.e., utility) and privacy performance behave as continuous functions, as the data's information content is removed or obfuscated, privacy will monotonically increase while the utility monotonically decreases.



(a) Classic Utility and Privacy Tradeoff

(b) Other Utility and Privacy Tradeoffs

Figure 1: UP: Relationship, between privacy and utility as a function of data. *Note that the shape of the curve on the left is just illustrative of a Pareto efficiency frontier. Not all Utility and Privacy tradeoffs may behave like this. Another example is the shape on the right, where there may be discontinuities.*

The quantity of information can be reduced by lowering collection frequency and resolution (e.g., lower camera resolution) or by privacy-preserving functions that actively transform the data to minimize or remove information content. Typically, data utility is lower with global data transformations (e.g., blurring an image) and higher with localized object protection (i.e., destroying the specific sensitive elements on the data that is independent of utility) [20]. Also, utility is often inverse to privacy (i.e., utility decreases, and privacy risks are mitigated as less data is shared). However, in practice is hard to predict how an obfuscation technique will affect both privacy and utility. The obfuscation may not increase privacy at all or suddenly increase privacy. The tradeoff can take any shape and may need to be defined with a step function as presented in Figure 1-(b). In practice, privacy solutions need to be evaluated empirically.

Another important factor in IoT systems is energy consumption. IoT deployments often have battery-operation requirements to accommodate deployment mobility, renewable-energy-based systems (e.g., solar panels), or temporary systems [82, 47, 44]. Also, beyond adding resilience against power failures, battery-operated setups can reduce the need for

electricians, minimizing installation labor costs while facilitating and expanding the number of device placement locations [32]. However, sophisticated privacy-preserving functions often need high computational resources and can impact the device's battery lifetime. Therefore, the energy required by these functions needs to be carefully considered in addition to the UP trade-offs.

These insights emphasize the need to make privacy, utility, and energy consumption a first order concern for IoT systems. IoT will need improved mechanisms for bolstering privacy with particular emphasis on privacy solutions deployed on-device while accounting for resource-constrained environments. This endeavor necessitates a comprehensive exploration and understanding of the unique challenges posed by privacy in the IoT context, thereby setting the stage for the impending discourse on the dissertation.

1.2 Thesis Statement

The goal of this dissertation is to justify the following thesis statement:

> *It is possible to design privacy tools specifically for Internet of Things (IoT) based resource-constrained devices, emphasizing mechanisms that enable local processing to achieve a delicate balance between privacy protection, utility preservation, and energy consumption.*

This dissertation justifies the above statement by proposing and solving the following three research questions:

RQ1 How can the characteristics of Utility, Privacy, and Energy (UPE) tradeoffs be understood and evaluated within the context of privacy-preserving functions in IoT devices?

RQ2 What methods can be formulated to maintain both privacy and utility from IoT devices, specifically in a setting that utilizes federated learning?

RQ3 How to selectively obfuscate sensitive data features while keeping utility information intact and minimizing the computing requirements for resource-constrained devices?

Each research question is explored in a chapter. *RQ1* is answered in Chapter 3, *RQ2* is studied in Chapter 4, and *RQ3* in Chapter 5. We now discuss the related research challenges.

1.2.1 Research Challenges

Below we list the research challenges related to the thesis statement and research questions.

1. **Development of Characterization Tools with new Capabilities**: Evaluating the impact of privacy-preserving solutions can be challenging because it is often difficult to predict how any given data transformation will affect specific utility and privacy requirements (which can vary depending on applications and users). Moreover, different data transformations can have different parameters to tune the intensity (e.g., the number of iterations in a privacy-preserving algorithm). Hence, frameworks and guidelines can be used to properly systematize the evaluation of privacy solutions to optimize their use, including identifying their energy requirements, understanding their impact on data utility, and evaluating their efficacy in preserving privacy. *Developing tools with the new capability of balancing Energy along with Utility and Privacy* while enabling comprehensive testing and characterization of *privacy-preserving functions* (privatizers) is a significant challenge, since they have to be generic enough to encompass different data modalities, privacy solutions, and devices in the context of IoT environments.
2. **Utility-Preserving Privacy**: A central challenge in this realm is the ability to perform data transformations on the data collected in a manner that preserves the utility of the data while still maintaining user privacy. Typically, any data modification intending to protect privacy will degrade its utility. The transformation must maintain the quality of the data for its intended purpose. Hence, developing transformation methods that can strike a balance between these two competing requirements is challenging, and it requires a comprehensive understanding of how gains in privacy incur losses in the utility of the data.
3. **Privacy-Preserving Machine Learning**: The advent of machine learning techniques in IoT brings about the challenge of ensuring privacy within these models. Given the

complexity of machine learning models (e.g., deep neural networks) and their ability to capture deep relationships in the data, there is an increased risk of revealing sensitive information for any data the user shares. The challenge here lies in developing privacy-preserving solutions that allow machine learning models to learn useful patterns from the data without capturing or revealing sensitive information. This also involves implementing techniques, such as differential privacy or secure multi-party computation, within the machine learning processes, which presents its own technical challenges. Furthermore, modifying the data in any way for privacy preservation can have unpredictable effects on the performance of machine learning models. Therefore, understanding the interplay between data transformation techniques and machine learning algorithms is crucial. The challenge is developing transformation techniques to obfuscate sensitive inferences based on ML while keeping the data suitable for other machine learning tasks.

4. **Sensitive Feature Consideration**: Clearly, not all data contains private information, enabling the possibility of applying the privacy-preserving method to only a subset of the data. However, understanding to what degree different portions of the data are related to sensitive information and creating mechanisms that only target these sensitive contents while maintaining the rest of the data intact is extremely challenging.
5. **Constrained Devices**: IoT devices are characterized by having limited memory, computing, and networking capabilities. Also, IoT devices often need mobility and flexibility for deployment required battery operation or energy harvesting. Hence, hence the execution of a privacy solution has to be tuned to IoT environments in order to be practical.

1.3 Overview of Dissertation Work

This dissertation will explore and propose utility-aware, privacy-preserving, and power-efficient data transformations in the IoT domain. The focus is to refine the understanding of the underlying characteristics of privatizers in IoT applications and their trade-offs to provide more efficient and secure systems regarding privacy, power consumption, and its effects in applications for future IoT systems. I will examine and develop solutions to the

challenges mentioned above by (i) developing and evaluating a theoretical model for the Utility, Privacy, and Energy trade-off of privatizers; (ii) creating a framework to guide the selection of privatizers; (iii) proposing an evaluation methodology to applications with Differential Privacy in Federated Learning; and (iv) proposing a power-efficient solution for the Utility-Privacy trade-off problem for IoT applications.

We start by discussing some background and related work in Chapter 2. Mainly, we review other privacy-preserving solutions and set up key concepts such as Privacy Enhancing Technologies, Privatizers, and data secondary use thread model. We also discuss prior work on balancing utility and privacy, privacy metrics, and specific privacy solutions for voice data.

In Chapter 3, we lay out the foundational work to answer the thesis statement and focus on research question *RQ1*. We propose a model for balancing the utility, privacy, and energy (UPE) and enabling analysis for IoT devices' energy needs while ensuring adequate privacy and utility. Our framework aims to guide users toward energy reduction while maintaining UP, choosing which privatizers to run locally (on-device). We address the following questions: (i) Can we model and design IoT privacy-preserving systems while considering energy? (ii) Do energy-efficiency considerations affect the utility-privacy tradeoffs? Further, this chapter explores essential considerations for defining privacy in the context of AI-powered inferences on sensitive data, particularly from neural network models. This research has been dedicated to formulating and evaluating privatizers that can successfully deter sophisticated privacy breaches in IoT systems.

In recent years, federated learning has emerged as a solution that mitigates the problem of sharing raw data by training models in a decentralized manner. In Chapter 4, we explore differentially private federated learning (DPFL) and study its effectiveness in mitigating privacy risks for energy-efficiency applications. Specifically, we look at the privacy challenges of smart meters to support *RQ2*. Smart meters are a class of IoT devices that can be used in non-intrusive load monitoring (NILM) applications, that is, home energy usage. In particular, we propose and evaluate how differential privacy can guarantee privacy in a federated learning setting. While home energy consumption data has many practical applications that can improve energy efficiency, it also leaks private information, such as user

behavior and occupancy. Such privacy concerns may prevent users from using smart meters, fearful of sharing raw energy data. In recent years, federated learning has emerged as a solution that mitigates the problem of sharing raw data by training models in a decentralized manner. In particular, we study the effectiveness of the DPFL in preventing an attacker from discerning the user participation in the training dataset (Privacy) while allowing accurate NILM predictions (Utility).

In Chapter 5, we propose a method to enable the privacy-preserving function to specifically target the sensitive contents of a dataset and support *RQ3*. In the previous chapters, we used blunt methods where all the data is obfuscated to remove or disrupt sensitive information. The next step is to find a method to dynamically isolate the sensitive features of a dataset and target an obfuscation mechanism to only those features. To address this need, we developed *PrivSpeech*, a novel, customizable framework for preserving privacy in IoT data. PrivSpeech can selectively obfuscate user-defined privacy-sensitive features, balancing data utility and privacy before exposing data to external entities. We focus on the human voice, a particular form of data with rich, sensitive information that has become commonly used in voice-based user interfaces (VUIs). By obfuscating selected privacy-sensitive voice attributes, the PrivSpeech model is also smaller, minimizing the computing requirements for data obfuscation and saving energy consumption. PrivSpeech is also further optimized to execute on constrained devices with minimal power footprint. In extensive experimental evaluations on various audio databases, we show PrivSpeech achieved an efficient and delicate balance between privacy protection and data utility, significantly improving the performance of both aspects when compared to the mere removal of sensitive features.

Depicted in Figure 2 is a conceptual overview of the research trajectory, partitioned into three key phases to support the thesis by answering the research questions. Below, we describe in more details each of the phases.

1. **Phase 1: Modeling Utility Privacy and Energy**

   The initial phase is dedicated to constructing the foundational UPE model, an augmentation to the conventional Utility-Privacy problem, incorporating a dimension of energy consumption. Also, evaluate the UPE model in two machine learning tasks with different

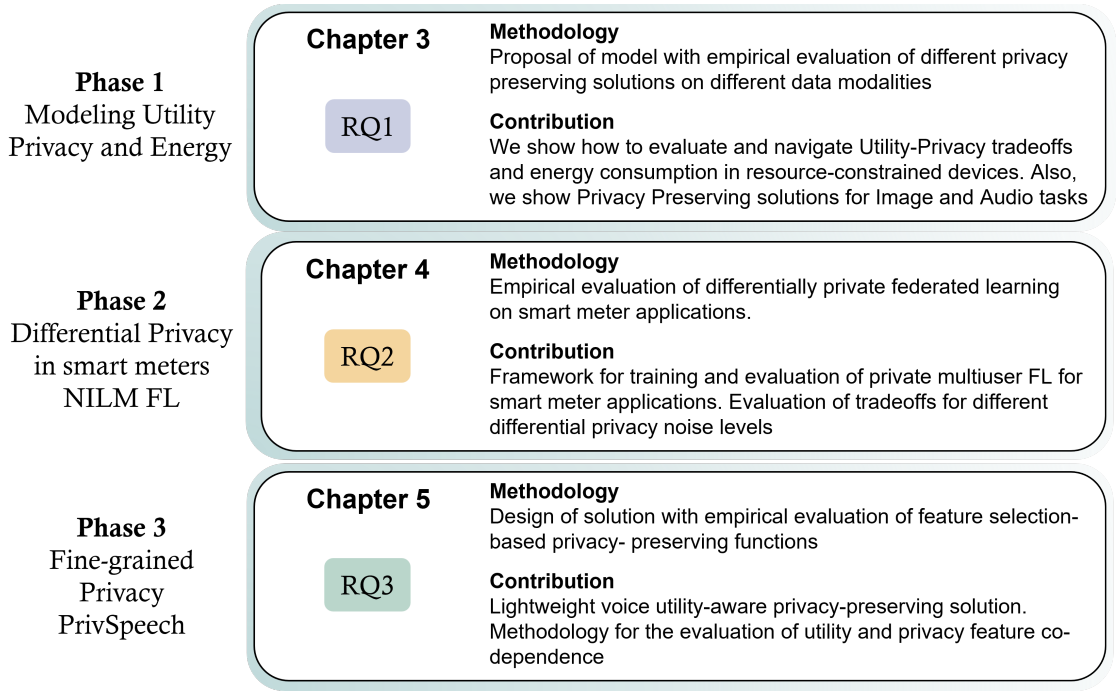


Figure 2: Research Trajectory overview

data modalities, image classification, and voice-to-text applications. We analyze several data transformation techniques as privacy-preserving solutions for each task and analyze their tradeoffs.

2. **Phase 2: Differential Privacy on Non-Intrusive Load Monitoring with Smart Meter data**

   In the second phase, the study's scope pivots to exploring a differential privacy solution within a federated learning environment specifically applied to an IoT device - smart meters. In this phase, we explore the role of differential privacy in federated learning for smart meters in the context of non-intrusive-load-monitoring applications. We trained a neural network model with differentially-private federated learning and analyzed the privacy tradeoffs with different metrics.

3. **Phase 3: Fine-grained Privacy - PrivSpeech**

   Finally, the third stage is characterized by an extensive refinement of the Utility and Pri-

vacy problem, wherein a solution is proposed to selectively address the sensitive segments of data, with a particular emphasis on human voice data. We design PrivSpeech, a framework composed of a lightweight neural network, PrivSpeechNet, capable of obfuscating private attributes while restoring the utility of the users' applications. We evaluate its performance on three datasets with three tasks, gender identification, emotion detection, and speaker verification. PrivSpeech model is further optimized to reduce computing and memory requirements to save energy consumption. We analyze the performance of PrivSpeech under varying combinations of tasks as utility and privacy (e.g., emotion detection as a utility while other tasks are private).

1.4 Contributions

In this dissertation, we tackle the challenging goal of balancing utility, privacy, and energy (UPE) efficiency in the context of Internet of Things (IoT) devices. This goal requires addressing several complex dimensions simultaneously, necessitating careful design, implementation, and evaluation of various frameworks and models. Our work leverages unique strategies to preserve privacy, ensure utility, and optimize energy consumption, which is critical for the efficient operation of IoT devices. Each of the ensuing contributions has been designed and implemented, with thorough evaluations and analyses conducted to substantiate our findings. These contributions collectively pave the way toward more sustainable and private IoT solutions.

1. **Design, implementation, and evaluation of the UPE Model**: We developed a novel model to harmonize Utility, Privacy, and Energy (UPE) considerations within IoT systems. Our model directs users to viable strategies for reducing energy consumption while maintaining a robust balance of privacy and utility. We have implemented and assessed our model within the image and audio tasks and different privatizers, demonstrating the successful identification and application of efficient privatizers for each task. Finally, we show an in-depth analysis of UPE tradeoffs that reveals their nonlinear characteristics as hyperparameters of privatizers vary. This analysis emphasizes the complexity

of selecting optimal privatizers.

- **Evaluation Methodology of Privatizers**: We introduced a method for evaluating image and audio privatizers and validated it across two case studies using the proposed UPE model. Our methodology aids in understanding the utility-privacy tradeoffs while distinguishing energy-efficient privatizers. The proposed model enabled the selection of better privatizers by identifying candidates with similar UP tradeoffs but less energy consumption for the image classification task. For the Audio modality, we designed a simple privatizer inspired by the evaluation of the image algorithms that showed the best performance along the UPE tradeoffs. Moreover, we highlight that the leading cause of the energy consumption of privacy solutions is not always the computation complexity of the privatizer algorithm but often related to the duration of device awake states.

2. **Development and evaluation of the DPFL Framework**: The creation of a differentially-private federated learning (DPFL) framework to train Non-Intrusive Load Monitoring (NILM) models. This framework offers a privacy-preserving distributed learning system that effectively mitigates privacy attacks. The DPFL framework was implemented as open-source modules, with integration capabilities for privacy attacks to measure DPFL's effectiveness in preserving privacy. This includes developing interfaces that allow extensions to existing NILM models and datasets.
    - **Evaluation of NILM Neural Network Models**: An evaluation of the different NILM models within the DPFL framework, providing insights into how different models behave with DP noise. Our framework can be used to develop neural-network models that are more resilient to DP noise.
3. **Implementation of PrivSpeechNet Obfuscation for Constrained IoT devices**: The design of *PrivSpeech*, an obfuscation mechanism for voice utility and privacy tasks that strategically identifies and obfuscates only sensitive features while preserving the integrity of the remaining features towards providing high utility for users.
    - **Exploration of Feature Selection Strategies**: Exploring different top-k feature selection strategies to inform task-specific voice obfuscation algorithms. This explo-

ration leverages Shapley values to efficiently erase sensitive features from datasets, optimizing the privacy-utility balance.

These contributions represent significant advancements in balancing utility, privacy, and energy efficiency in IoT devices while providing critical knowledge and tools for further research.

2.0 Background

2.1 Concepts and definitions

2.1.1 Privacy

The concept of privacy, given its intricate nature, lacks a universally accepted measure [139]. The endeavor to ensure privacy in any system hinges on its multifaceted demands, encompassing subject-dependent and cultural values like human autonomy, dignity, and diversity in potential solution approaches. Privacy can manifest through several strategies, such as limiting access to sensed data, promoting isolation, or controlling information flow related to specific identity attributes [2].

In this dissertation we adopt the concept of IoT *Privacy Risks* as defined in [112]:

- **Secondary Use (SU)**: Collecting or using the data for purposes other than those initially consented by the data owners.
- **Unauthorized Access (UA)**: Breaching confidentiality during any data collection or transmission phase without proper authorization.

Note that unauthorized data access does not necessarily prevent secondary usage and vice versa (e.g., authorized service providers may use the data for other purposes, while data with no secondary uses does not prevent unauthorized access).

This dissertation defines a *privacy violation* as an unrequested data exchange with a known or unknown party, which can occur via a primary or a side channel. A violation in the primary channel may involve the leakage of sensitive data beyond what is necessary, such as images that inherently carry more information than required [112]. Conversely, a side-channel violation comprises the gathering and misuse of ostensibly unrelated data (for example, device logs) to deduce confidential information [112]. These violations collectively fall under 'Secondary Use'. *The primary focus of this dissertation is 'Secondary Use' in the context of primary communication channels with known attacks.*

In the context of 'Secondary data usage', it becomes essential that the stakeholders involved in data exchange uphold the responsibility of ensuring only the required or mutually agreed upon data is exchanged. In scenarios where additional data is being leaked, this must occur only with the knowledge and consent of the data source owner. Consequently, privacy solutions for the Internet of Things (IoT) need to address the following concerns:

- **User privacy requirements** can define the scope of applicable privacy solutions. Therefore, users should specify an agreed-upon boundary of use to the data they share, allowing developers to focus only on protecting what users consider sensitive. However, the average user, and sometimes even advanced ones, may not understand or anticipate the dangers of the data they are willing to share.
- **Application requirements** can be used to quantify and implement mechanisms that enforce the collection of the minimum amount of data needed to function. The user is likely to leak more information and be exposed to unknown risks if more data than necessary is sent (e.g., sends 4K image when only 2K was needed). This type of solution is typically enforced as a policy or a design guidelines [112, 104]. In practice, the data collection, processing, and its applications have to be simultaneously tuned, which can be challenging in the device heterogeneous environments expected from IoT systems [117].

Addressing user and application requirements is critical in mitigating the risks associated with 'Secondary data usage'. However, even with these requirements fulfilled, a considerable challenge remains: assuring that data, especially information-rich data such as unstructured, time-series data (such as audio and video), does not inadvertently disclose sensitive information. Modern machine learning techniques have shown increasing success in extracting information from seemingly innocuous data. Secondary data usage is mitigated when application and user requirements are addressed; yet, it can be hard to guarantees that the data does not contain any other extractable sensitive information.

2.1.2 IoT Systems' Architecture and Security

IoT devices are often bundles of sensors and resource-constrained hardware and rely on the Cloud for various data services and analytics. Figure 3 presents the typical IoT

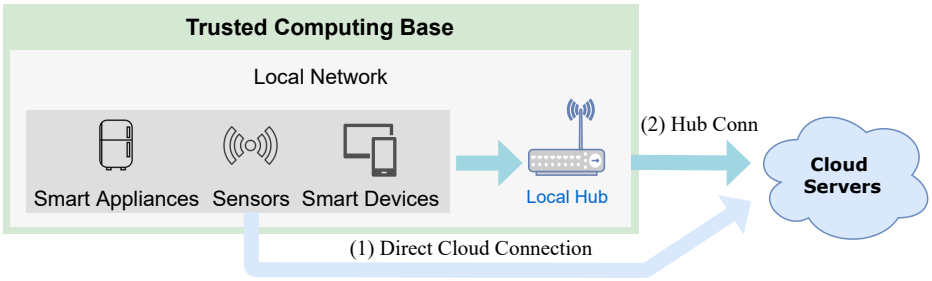


Figure 3: Cloud-based IoT systems follow a two or three-tier architecture TCB.

system, in which devices (e.g., thermometers, cameras, or voice assistants) send data to cloud services either directly or via a gateway hub, such as Samsung SmartThings [135] or Amazon Echo [57]. Even when there is a hub, raw sensor data can be processed in the Cloud for analytics or sent for storage, as hubs tend to have Raspberry-Pi-class processing and networking capabilities, insufficient for running large machine learning models.

While there have been recent efforts to run analytics locally, such as Google Home Assistant [91], which executes some speech recognition tasks locally, most deployments rely on cloud-based services and will continue to do so for the foreseeable future. A good example being the Generative Pre-trained Transformer 3 (GPT3) that has shown a significant increase in performance with a model that has 175 billion parameters and needs an HPC class specialized hardware to execute [31]. *Hence, I consider secondary data usage of intentionally, or not, shared data, as the threat addressed in this proposal.*

Conventional IT security is not prepared to face the challenges created by IoT devices [28]. IoT devices can have actuation capabilities that can interface with and modify physical systems. Simultaneously, IoT devices have operational requirements for performance, reliability, resilience, and safety often are at odds with common cybersecurity and privacy guidelines (e.g., higher resolution smart-cameras may improve a predictive model but may leak more information about users in its vicinity and increase the energy consumed) [28]. Moreover, IoT devices may need specialized software tools since they often have different vendors and manufacturers, which further add vulnerabilities due to integration challenges. The protection mechanisms of security and privacy for IoT devices can be defined in three high-level

goals:

1. **Device security.** Prevent device misuse and intrusion into an IoT network. This includes the protection of accessing the device's hardware, the impact on other devices if it is compromised (e.g., used to perform distributed denial of service), or eavesdropping into network traffic.
2. **Data Security.** Protect the Confidentiality, Integrity, and Availability (CIA) of the data itself that can be stored, collected, processed, or transmitted by an IoT device. At this level, cryptographic protocols can be used to protect many of these risks. The goal is to ensure controlled access by trusted parties.
3. **Data Privacy.** Protects the privacy of individuals contained in sensitive data that has to be processed by external services. This goal is outside the device or data security domain, requiring special different mechanisms to be addressed (e.g., privatizers).

Each security goal addresses different threats and builds on the previous, but it is orthogonal to each other. The security of devices' hardware defines its resistance against physical tampering, while data security ensures that private information is not accessible by unwanted parties as the data transits through networks. However, the actual contents of data and potential hidden features, especially in raw sensor data such as video and audio, often contain Personally Identifiable Information (PII) and profiling information of objects and people within sensor's vicinity. These threats are not completely covered by protection level 1 or 2.

This proposal focuses on data privacy threats (level 3). Raw sensor data can be a rich source of information with the potential to reveal seemingly uncorrelated details that often elude those to whom the data belongs. For example, researchers have previously shown how voice recordings can help diagnose medical conditions such as depression and schizophrenia [62], domestic abuse [125], or determine a person's mood [128, 133, 73]. Sensors such as gyroscope and accelerometers can detect when a person falls, useful in assisted living, or track medication routine and compliance [51, 114, 87, 89, 108]. Recent advancements in machine learning (ML) have made the detection of inconspicuous patterns on data possible, and newer methods are being developed to push the boundary of what we can infer from such

sensing data, such as diagnosing seizure disorders using smartphone-based electroencephalogram (EEG) [134, 9] or facial expressions [10, 11]. Unfortunately, not all ML models are developed for social good, and ML results may lead or facilitate burglary, stalking, physical aggression — a reason why there is a need for measures to prevent unrestricted analysis of raw sensing data.

Table 1: Examples of privacy threats and utility from sensing data in IoT environments.

Data Type	**Source**	**Utility**	**Privacy Threats**
Text	device usage logs and metadata	intrusion detection and usage history	device identification and user profiling
Audio	smart assistants	voice-based services	voice recognition and private attributes
Images & Video	smart cameras	threat detection	face recognition and people tracking
Other Sensors	accelerometer, gyroscope, and optical sensors	health monitoring	localization and daily routines behavior

Table 1 provides a sample of useful services and privacy threats that can be exploited from data generated by IoT devices. However, it may not be possible to constrain one privacy threat to one device or data type being collected; hence, different devices can share the same privacy threat. For example, a person could be identified, characterized, or have his/her location determined by audio, video. Other sensors such as temperature and humidity can also correlate with different events that could be extracted with ML techniques.

- *Textual Data*: text is often overlooked when considering privacy invasion, but it can be used to infer many properties from individuals. Text mining techniques leverage social network feeds, emails, twits, forum discussions, and exchanged messages, and apply Entity Recognition and Relation Extraction to create structured data [13]. Sentiment analysis techniques use text to derive the mood of individuals. *Audio*: with the growing popularity of smart voice assistants (e.g., Google Home and Alexa), and the voice recording leaks on both platforms, privacy related to audio has raised severe concerns [60, 132]. Voice recordings can be used to determine mood, extract information to provide products of interest, and even to support the diagnosis of some medical conditions such as

depression and schizophrenia [62]. Moreover, the pandemic from 2020 have accelerated the adoption of remote tools such as Videotelephony (e.g., Zoom) and MOOCs.

Video and Image: advances in ML techniques coupled with easy-to-use programming frameworks (e.g., TensorFlow) have opened vision analytics to almost anyone. Video is a feature-rich data type and can capture minute details from the environment. For instance, ML techniques can derive age, gender, ethnicity, face recognition, photoplethysmography (i.e., heart rate), breast cancer diagnosis, and more from images and videos [12]. Notably, previous work showed that it is possible to even extract sound from silent videos through Visual Vibrometry [38, 85].

- *Other sensors and actuators:* temperature sensors or actuator logs can contain a lot of information. These devices capture data about physical events through time (i.e., time-series data), which can be interlinked with other events in different manners. For example, smart-meter data allows attackers to profile users' daily routines and discover house appliance usage [19].For example, temperature data can correlate with other events such as the number of people in a room, the AC system, or opening room windows.

  Nevertheless, the type of attack itself can still have different data quality requirements. For instance, single-digit weekly power consumption can be enough to profile the target social-economic status, while decimal precision data with 1Hz collection frequency allow attackers to detect user's appliance daily usage.

IoT systems are directly at odds against privacy. While a successful IoT system requires frictionless connectivity and integration between devices and the broad Internet, privacy demands isolation and control. *Policies and guidelines* is one solution that pushes the privacy protection responsibility to product and software developers [112, 28]. However, the IoT market still largely unregulated. IoT applications will need theoretical and software support to address the UP trade-off's dynamic nature to address multi-user privacy requirements in shared spaces, added trade-offs from device power constraints (UPE), sensors that change the resolution and collection frequency, or the emergent patterns from combined devices' data. A more generic direction has to rely on software-assisted solutions. Privacy-preserving functions (*privatizers*) can be used to remove sensitive information and prevent secondary-usage attacks from inferring or extracting private information. There is a broad literature

that implements privacy solutions that could be applied in IoT and could be considered a privatizer [77, 64, 105, 63, 36, 28, 112]. Conventional privatizer approaches include data obfuscation functions [22, 49, 17], local differential privacy [45, 46, 98, 24]. For example, data obfuscation functions can be implemented with the addition of noise to data or blurred with convolutional matrices to thwart any privacy attacks [63, 35].

Utility and privacy are also often at opposing ends; increasing privacy through suppression usually destroys valuable information in the data, private or not. Hence, any analytics run on the privatized data may have degraded performance and can prevent valuable insights. However, some privatizers permit balancing this trade-off between privacy and utility, such that we can still achieve some utility for the user while reducing the ability of attackers to exploit the data. Information-theoretic models can be used to quantify the privacy leakage in lieu of the UP trade-offs with entropy-based metrics that measure the probability of information novelty or quantity for the attacker [139]. However, these metrics are not easily translated to a given attack's success rate (e.g., the correlation of an image's entropy with facial recognition model false positive rate). Moreover, entropy-based metrics ignore possible apriori-information used in an attack, which pre-trained ML models can capture.

2.1.3 Privacy-Enhancing Technologies and Privatizers in IoT

Privacy-Enhancing Technologies (PET) encompass privacy-protection methods, including encryption, access control mechanisms, privacy-by-design guidelines, and privacy-preserving techniques [29, 104]. For users relying on remote services requiring data access to deliver desired functionalities—such as access to unencrypted data for model training or inference—the primary line of defense is removing sensitive information before data transmission. This necessitates using privacy-preserving functions, or 'privatizers' as referred to in this proposal, which removes sensitive information from data. These 'privatizers' can employ several techniques, such as obfuscation (like blurring an image), data minimization (like feature extraction or compression), or even sophisticated deep neural net (DNN) models dedicated to sensitive information removal [65, 120].

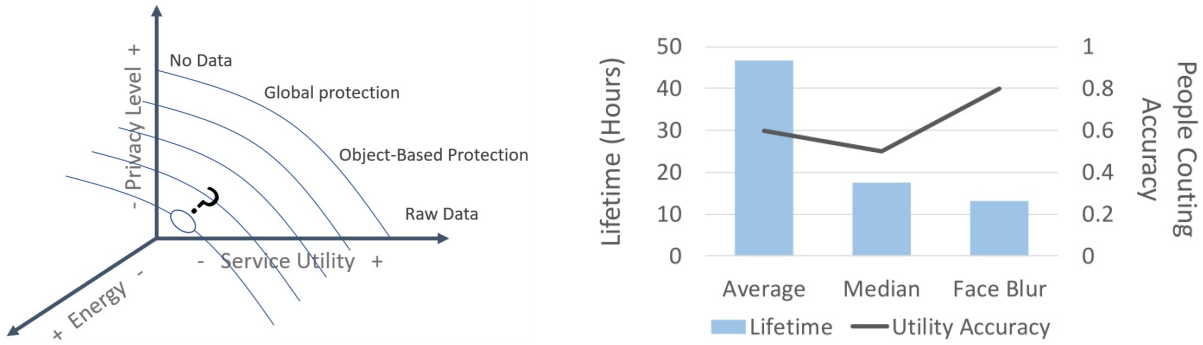
Privatizers. The mechanisms used to remove sensitive information from data or for anonymiza-

tion are often referred to as Privacy-preserving functions [61] or data sanitation techniques [106, 109]. In this dissertation, we extend the terminology used in [65] beyond autoencoders[1]: we refer to *IoT privatizers* as mechanisms used to remove private features from data [61]. Privatizers do not need to destroy information since modifying the data can still mitigate attackers with limited resources. Examples of privatizers include obfuscation (blurring an image), data minimization (e.g., extracting the main features and compression), text and image redaction, and autoencoders trained to remove sensitive information [65, 120]. We also adopt privatizers due to a recommendation of IoT *privacy by design* principles as the first line of defense against private information leaks [111].

The balance between the utility and privacy (UP) trade-offs of a 'privatizer' primarily hinges on the degree to which both aspects rely on the same data features. This can be particularly apparent with machine learning (ML) models, where the exact features learned (i.e., the patterns identified) are often complex and can vary based on the model or the data. Researchers commonly resort to experimental validation to comprehend the potential risks of data release and its utility for sensitive inference. Gaining insights into how 'privatizers' can eliminate sensitive patterns can inform more secure data-sharing solutions. The IoT context exacerbates this problem as each additional device potentially harbors sensitive data and can generate intricate patterns with other devices, consequently correlating further with sensitive information.

In IoT systems, the energy consumption of devices is a critical design aspect due to their inherent resource constraints and potential requirements for battery operation and mobility. Energy-performance and energy-aware-security models have previously been explored in general-purpose computing [86, 138]. However, attempts have yet to simultaneously model energy, privacy, and utility specifically for IoT systems. Although power consumption is often disregarded when considering the cost of privacy mechanisms (with organizations or individuals prioritizing privacy and going to great lengths to secure it), battery operation can be vital for IoT devices. This is due to the potential to simplify deployment and reduce associated costs. Consequently, studying the trade-offs of 'privatizers' could empower designers to navigate the landscape of potential software-based privacy solutions effectively.

[1]Autoencoder is a type of Neural Net used to learn a compressed representation of data.



(a) UPE *pareto frontier* (adapted from [141])

(b) Utility and Power Consumption trade-offs for three different image privatizers

Figure 4: (a) UPE: Complexity increase with the addition of energy to the classic UP problem; (b) The energy and utility trade-offs of different image privatizers

Figure 4-(a) illustrates the addition of energy consumption that extends the choice for possible privatizers in the Utility-Privacy-Energy trade-off space and Figure 4-(b) shows the Utility/Power consumption trade-offs for three simple image privatizers; two are performing a global blurring on the image with an average and median kernels while the last is a face-localized blurring to protect a person's identity against face recognition. This illustrates that the choice of a privatizer can depend on non-functional requirements beyond the classic UP trade-offs.

The challenges of preserving privacy in IoT systems are multifaceted, involving utility, privacy, and energy consumption trade-offs. As Figure 4 illustrates, various factors beyond the classic Utility-Privacy dichotomy can influence the choice of privatizers and the resultant trade-offs in an IoT environment. Particularly in energy-constrained IoT systems, exploring energy-efficient privacy solutions becomes a significant concern. By studying these trade-offs and understanding the performance of different privatizers, we can move towards designing more effective, energy-efficient, and privacy-preserving strategies tailored to the unique requirements of IoT systems. A continued exploration of privacy-enhancing technologies and the deployment of efficient privatizers will play an integral role in ensuring privacy while leveraging IoT systems' applications potentials.

2.1.4 Energy consumption in IoT systems

IoT devices are often deployed with battery-operated setups to enable applications of mobile, renewable-energy-based (e.g., solar panels), or temporary systems that require minimal user maintenance [48]. The added resilience against power failures and battery-operated setups can also obviate the need for electricians, reducing installation labor costs while giving freedom to the location where the setup can be mounted/installed [32]. In battery deployments, reducing maintenance frequency (e.g., charging or battery replacements) is important [83].

The privatizers will require additional computational resources, increasing the overall energy footprint. As we highlighted in Section 2.1.2, the need for close-proximity data processing, essentially executing potentially heavy computations on IoT sensors themselves, is imperative for ensuring privacy. While critical for privacy, this requirement could present a challenge for IoT deployments designed for mobility or powered by renewable energy, where resource constraints are particularly pronounced.

For example, assume a mobile battery-operated system that uses a camera to count people in different events, and the user is evaluating two privatizers for protecting the identity of people; privatizer 1 consumes 4KJ more than privatizer 2. Also, consider that there are three sessions a day, four days a week, which translates to 12 sessions a week. With a reference battery of 298WH (typical in portable power stations), Privatizer 2 will last 11 weeks compared to 7 weeks of Privatizer 1, thus lasting almost a month longer (assuming no idle discharge).

In light of these constraints, the energy consumption associated with the execution of privatizers becomes a significant factor when evaluating their suitability in IoT devices. Many IoT devices, particularly those intended for mobility or dependent on renewable energy, operate under strict power constraints. Running resource-heavy privatizers could rapidly deplete energy reserves, impacting the system's mobility, operational longevity, and overall efficiency.

Consequently, it is essential to develop frameworks capable of identifying energy-efficient privatizers for these energy-constrained devices. Such a framework should provide a com-

prehensive view of utility, privacy, and energy trade-offs. This approach would facilitate informed decision-making in selecting privatizers, taking into account their effectiveness in preserving privacy and maintaining data utility and their alignment with the energy limitations inherent to mobile and renewable energy-powered IoT systems.

2.1.5 The Inherent IoT Privacy Risks

IoT data is especially prone to secondary usage as sensed data can contain information that has many applications. For example, audio data from smart assistants may reveal personal information (e.g., age, emotion) [81, 103]. A user may be comfortable sharing the data for a particular purpose but not for secondary purposes.

This can be explained by observing a fundamental consequence of the widespread use of IoT sensors. Humans use different senses, shaped by evolution, to perceive the patterns of the physical world. These senses evolved only to perceive what was essential to allow the survival of the human species and much of the physical world is not perceived. However, we have extended this perception through technology, allowing us to see and hear beyond the visible electromagnetic and audible sound wave spectra. The boundaries of our ability to detect patterns have been further extended in the information age, which has brought ever-growing data, prompting advances in the fields of Statistics and Computer Science that enabled the extraction of usable information by detecting hidden correlations and causal relationships in data from different sources.

Sensor data and the information it can provide relies on the assumption that the data is "well-behaved" and contains patterns that corroborate the phenomena it aims to describe or measure. For example, the patterns transmitted by light and sound waves are"well-behaved", allowing animals to perceive and create languages through the relationship of distinguishable phonemes and graphical symbols. However, the range of patterns in data distinguishes how much information it may contain. Data sources that generate a random or constant signal cannot provide any value and would be indistinguishable from noise without context (e.g., temperature measurements may look random if the sampling rate is unknown). However, physical events can be interlinked in ways that often elude human perception. The simple

measurements of a temperature sensor can be correlated with various other events, such as the number of people in a room, the AC system, or if the windows are open.

IoT devices (sensors and actuators) typically have computing components and networking capabilities that allow sharing data or receiving commands through the Internet. As a physical device, it collects data with two intrinsic properties: (i) the data can always be associated with a location (i.e., geographical location), and (ii) each data point has a relation with time through the sensor's collection frequency. Hence, the type of data that a sensor generates typically is a time-series (*cf.* cross-sectional or one-shot data) associated with a location. In other words, sensors capture the history of change in physical phenomena. They can reveal much more than what they were designed for, capturing minute and humanly undetectable patterns (micro-signatures) that together can compose a fingerprint (a set of micro-signatures) that highly correlate with other properties of the environment or those in the device's vicinity (i.e., bystanders).

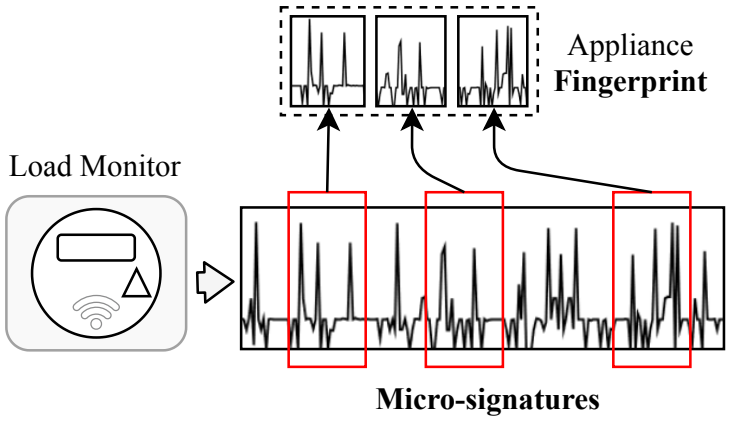


Figure 5: Data micro-signatures and fingerprints.

Figure 5 shows how fingerprints can be composed of micro-signatures on load monitoring data (i.e., electricity usage in a house). Appliances have different energy consumption that depends on the appliance model or brand but can be distinct enough to be recognized (e.g., fridge). Non-intrusive load monitoring techniques can be used to disaggregate an appliance fingerprint signal from whole-house energy consumption data [26]. The fingerprint of appliances can share similarities (i.e., micro-signatures) caused by appliances having similar

components (e.g., peak load when turning on) or their usage routine. This principle can be applied to images, audio, or sensed data. Machine Learning applications' success relies on the assumption that the data contains fingerprints correlating with the desired prediction goal. There is no need to formulate the shape or any property of fingerprints. A Deep Neural Net model can learn from any arbitrary pattern and even memorize data-label relationships if no pattern exists [148, 18].

2.1.6 Federate Learning and Differential Privacy

Federated Learning (FL) coupled with Differential Privacy (DP) offers a promising solution to privacy concerns, particularly in IoT settings. FL techniques, which perform a portion of an application's computation task on the user's local devices, allow machine learning (ML) models to be trained locally, thus eliminating the need for users to expose their raw data [96, 33]. This substantially reduces privacy risks associated with transferring user data to a centralized server. Despite these advantages, FL alone may still inadvertently leak data about the training set, making it susceptible to malicious attacks such as membership inference [71, 144]. To mitigate this vulnerability, DP has been integrated into FL. DP is an established industry technique that enables the aggregate analysis of multi-user datasets without revealing the participation of any specific user in any given output [46, 24].

Let D and D' be two datasets that only differ by one element (i.e., one user). DP specifies that for an algorithm A that performs an aggregate analysis, the output probability of A for D and D' is bounded by the multiplicative factor e^{ϵ} (Equation 1).

$$Pr[A(D_1) \in S] \leq e^{\epsilon} * Pr[A(D_2) \in S] \tag{1}$$

A is assumed to have a mechanism to randomize its outputs. S is the set of truth answers without a randomized mechanism on A. The parameter ϵ is used to measure the similarity between the probabilities Pr for $A(D_1) \in S$ and $A(D_2) \in S$. Hence, DP addresses a specific threat model in which the attacker wants to identify if specific users participated in A output. DP is used to protect against membership inference.

Still, parameter ϵ from the DP frameworks does not capture the existence of hidden

features or measure if unique attributes can be inferred from an exposed model or raw data [8]. DP refers to an attacker's ability to infer if a particular data sample has participated in specific aggregate analytics. A randomization mechanism is added to the function' output such that the probability of the record that creates the highest variance in the query's result (which would make a record distinguishable) is minimized. Hence, the DP threat model targets information leakage on the function (i.e., the utility's output) and not on the risk of data secondary use.

3.0 Exploring Utility, Privacy, and Energy Tradeoffs: Characterization and Evaluation of Privacy Functions in IoT Systems

3.1 Introduction

The Internet of Things (IoT) has created new opportunities for data collection and analysis but also presents challenges in maintaining user privacy. As shown in Section 2.1.2, IoT data is transmitted to remote cloud services for processing and inference, and in this context, users have limited control over the extent of information extracted from their data. Also, building on the discussion in Section 2.1.3, recent trends propose addressing these privacy concerns through local obfuscation of data, executed either on-device or via an IoT hub/gateway. This strategy employs privatizers — privacy-preserving functions -– to obfuscate the data prior to sharing it with external services. An example could be smart IoT cameras blurring faces in images before transferring them to the cloud for further processing.

This obfuscation process can inadvertently diminish data utility since it might also remove useful information, thus limiting the potential applications that rely on the data. The trade-off between utility and privacy (UP) has been the central issue in numerous studies. However, as discussed in Section 2.1.4, energy is an often-overlooked aspect of this dynamic but essential given to IoT devices. The energy expenditure of privatizers can swiftly drain the limited energy resources of IoT devices, affecting the broader usability of such systems.

This chapter introduces a novel model that encapsulates the interplay between utility, privacy, and energy (UPE), considering the constraints typical of IoT environments. We aim to demonstrate that it is feasible to factor in the energy requirements of IoT devices while preserving privacy and keeping the data useful. Our framework is intended to assist users in minimizing energy consumption while maintaining the UP balance and deciding which privatizers to deploy locally on their devices. This exploration is guided by two primary questions: (i) Can we model and design privacy-preserving IoT systems with energy considerations in mind? and (ii) How does incorporating energy-efficiency considerations impact the traditional utility-privacy trade-offs?

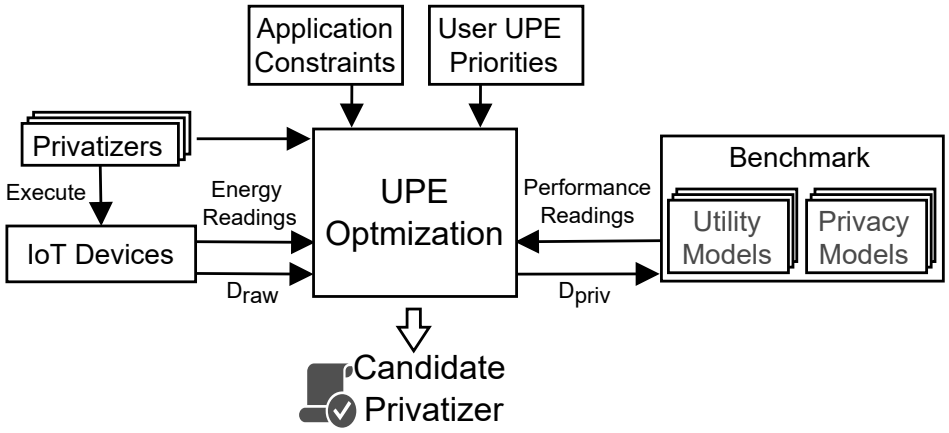


Figure 6: UPE tradeoff framework

3.2 Utility, Privacy, and Energy Framework

We develop a framework to carefully examine the cost of executing privatizers on devices. A key aspect is identifying energy-efficient privatizers that meet user and application demands while addressing UP tradeoffs.

Figure 6 presents the components of our UPE framework. The central module is the UPE optimization that computes the UPE tradeoffs for each privatizer based on: (i) how much energy the device consumes to privatize the data D_{raw}; (ii) the applications performance on privatized data D_{priv}; (iii) the performance of *known* privacy attacks on D_{priv}; and (iv) user priorities and application constraints. Based on these constraints, our UPE model finds candidate privatizers.

First, we define a set S, where $s \in S$, of privatizers such that $s(D_{raw}) = D_{priv}$. Then, each UPE objective for a particular privatizer s is specified as follows. The utility is measured as utility-loss $U(s)$ that estimates how much the privatizer degrades the performance of the user's services. Similarly, we can specify the privacy model as privacy loss $P(s)$, which measures how well attackers extract user-sensitive contents from the data. Finally, we define the energy loss $E(s)$, the energy costs for executing privatizer s in an IoT device. The framework allows a user (e.g., the system manager or developer interested in protecting the user's privacy) to specify multiple utility and attack models (i.e., related to private

inferences). These privacy and utility specifications must be done before the model identifies the best privatizer for the application.

We note that utility and privacy models can differ for different tasks and the metrics that quantify the performance of these tasks depend on the application. For example, the application utility can be specified as a regression or classification task. Similarly, performance metrics may also have a different interpretation depending on the value. For instance, some performance metrics (e.g., accuracy) indicate better performance for larger values, whereas others (e.g., error rate) indicate higher performance for smaller values.

Thus, to generalize our UPE model for different application scenarios and accommodate different metrics, we normalize the application's performance metrics using a min-max function. Let performance metric $m \in M$ denote the normalized performance metric such that its value varies between 0 and 1 (i.e., $0 \leq m \leq 1$), and higher values indicate better performance. However, in some cases lower values of the normalized performance metric m' (e.g., an error rate) will indicate better performance. To ensure that higher values indicate better performance, we can modify the metric as $m = 1 - m'$. In what follows, we assume that $U(s), P(s)$, and $E(s)$ values lie between 0 and 1, and higher values indicate better performance.

3.2.1 Utility model

We assume that the user will have multiple utility performance metrics F to capture the utility of the data. Let $f \in F$ where $F \subseteq M$, denote the utility performance metrics that computes the performance on the raw D_{raw} or privatized data D_{priv}. We define utility loss u for privatizer s as:

$$u(s) = 1 - f(D_{priv})/f(D_{raw}) \tag{2}$$

where $0 < f(.) \leq 1$. Then, we aggregate the utility loss of a privatizer $U(s)$ across all utility metrics for a single application (task) as follows.

$$U(s) = \sum_{i=1}^{|F|} \mu_i \times u_i \tag{3}$$

where u_i is the utility loss for metric i and μ_i is relative importance of metric i such that $\sum_i \mu_i = 1$. We note that $U(s) = 0$ represents no loss in utility for the application, and we are interested in minimizing this metric.

3.2.2 Privacy model

As above, we assume the data needs to be protected against multiple privacy attacks A. Let $a \in A$ where $A \subseteq M$, define the set of metrics that measure privacy attacks. Then, we define privacy loss p of a privatizer s as follows.

$$p(s) = a(D_{priv})/a(D_{raw}) \tag{4}$$

where $a(D_{priv}) \leq a(D_{raw})$ and $0 < a(\cdot) \leq 1$. We note that a lower privacy loss p indicates success in mitigating privacy attacks by the privatizer. Moreover, we compute the aggregate privacy loss $P(s)$ as:

$$P(s) = \sum_{j=1}^{|A|} \tau_j \times p_j \tag{5}$$

where p_j represents the privacy loss for metric j and τ_j is the relative weight of metric j such that $\sum_j \tau_j = 1$. As before, lower values indicates a better privatizer performance in mitigating privacy attacks across all user-defined privacy metrics.

3.2.3 Modeling Privatizer Energy Consumption

We measure the energy consumption of the privatizer and the device using the model in Figure 7 and Table 2 presents the stages and symbols used by each stage. As shown, we assume that energy consumption can be divided into three key stages: (i) sensing (sen), when sensors are active, (ii) privatizing ($priv$), when the privatizer executes, and (ii) network communication (net), when the data is transmitted to the Cloud. We note that dividing the energy consumption into distinct stages helps us approximate the energy footprint of the privatizer in the IoT device.

- **Device (base) energy** is the base amount of energy consumed by the device to maintain a ready state (da).

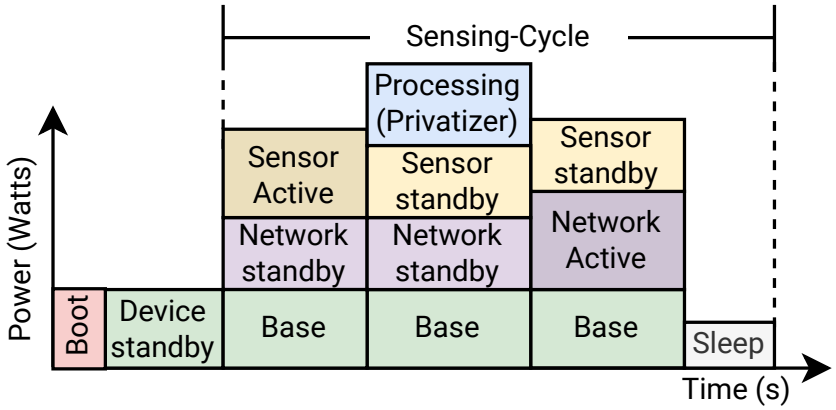


Figure 7: Energy consumption of an IoT sensor

- **Sensor Energy** is the energy consumed by the sensor when in standby (ss) or active (sa). We note that the different sensor hyperparameters (e.g., frame rate or image resolution of camera) may affect the energy footprint.
- **Network Energy** is the energy consumed when in standby (ns) or actively sending data (na). The energy used by the network depends on data size, which may be modified by the privatizer (e.g., when removing sensitive contents or adding noisy data).
- **Privatizer Energy** is the energy consumed by executing the privatizer (pr) and depends on several factors, such as the privatizing algorithm, execution time, and the underlying hardware (e.g., multi- or single-core devices).

Having different components facilitates the characterization of the total energy consumption. For example, we can isolate and measure the current of each component, which is the difference between the total current (e.g., using the sensor) and the base current during standby mode. Similarly, we can activate the network to measure the current in active or standby mode. Using the measured current, we can then calculate the energy consumption e as the product of voltage V, current I and time t, i.e., $e = V \cdot I \cdot t$. We use this energy equation to model the energy consumption of each stage. In particular, the energy consumed in the sensing stage e_{sen} is defined as:

$$e_{sen} = V \cdot (I_{sa} + I_{ns} + I_{da}) \cdot t(sen)$$

Table 2: Energy model variable description

Var	Description
sa	Sensors active current (Ampere)
ss	Sensors standby current (Ampere)
na	Network active current (Ampere)
ns	Network standby current (Ampere)
pr	Privatizer execution current (Ampere)
da	Device standby/active current (Ampere)
t	Sustained current duration (seconds)
v	Nominal device voltage (Volts)

where I_{sa} is the current when the sensor is active, I_{ns} represents the current when the network is in standby, I_{da} is the base current to keep the device active, and finally, $t(sen)$ is the duration of the sensing stage. Similarly, we define the energy consumption of the privatizer e_{priv} and network communication e_{net} as:

$$e_{priv} = V \cdot (I_{pr} + I_{ns} + I_{ss} + I_{da}) \cdot t(priv)$$
$$e_{net} = V \cdot (I_{na} + I_{ss} + I_{da}) \cdot t(net)$$

where I_{pr} is the current for the privatizer component, I_{ss} is the current for a sensor in standby, I_{na} is the current for the network in use, $t(priv)$ and $t(net)$ are the duration of the privatizing and network communication stages. Finally, we can compute the overall energy footprint $E(s)$ of executing a privatizer s.

$$E(s) = n(e_{sen} + e_{priv} + e_{net}) \tag{6}$$

where n is a normalization function such that $E(s) \in M$.

3.2.4 Optimizing for Utility, Privacy, and Energy

We now describe how we incorporate the above utility, privacy, and energy models into our framework. We first consider the user constraints to filter the set of candidate privatizers that meet the minimum requirements. For instance, the user may require a minimum utility from the data. We capture this using a threshold-based approach. In particular, the set of candidate privatizers L is defined as:

$$L = \{s \mid U(s) \leq c_1 \wedge P(s) \leq c_2 \wedge E(s) \leq c_3, \forall s \in S\} \tag{7}$$

where c_1, c_2, c_3 are the thresholds for utility, privacy and energy. For example, when E is bounded by a constraint c_3, we mean that the overall energy consumption of using the privatizer cannot exceed c_3. Once the threshold constraints are satisfied, we then use an optimization model to meet the UPE objectives defined by the user. We define the overall UPE objectives as the sum of utility, privacy, and energy losses.

$$\arg\min_{s} \; G(s) = \alpha U(s) + \omega P(s) + \sigma E(s) \tag{8}$$

where $s \in L$, and α, ω and σ are hyperparameters that allow user to control the importance of each objective when identifying the ideal privatizer. Further, we assume $\alpha, \omega, \sigma \in [0, 1]$ and $\alpha + \omega + \sigma = 1$.

When $\alpha = \omega = \sigma$, all factors have the same importance, and setting a particular weight to zero is the same as ignoring the corresponding objective. For instance, if the user values energy twice as much as utility and privacy, we can set the importance weights as $\alpha = 0.25$, $\omega = 0.25$, and $\sigma = 0.5$. Further, when $G(s) = \epsilon$, where ϵ is an arbitrarily small positive quantity, it means there is a negligible loss in the utility of the data, the privatizer s was able to successfully mitigate the attack, and the energy consumed by the privatizer s was negligible.

By minimizing $G(s)$, the user can find the best privatizer that maximizes utility and privacy while minimizing energy loss. Hence, Eq. 8 can always lead to the smallest loss based on user-defined weights to prioritize utilities, privacy, or energy. Note that each UPE objective function has no unit, since they are computed as ratios.

3.2.5 Using the Model

The following steps guide the formulation of the UPE model, that is, what are the steps when a user wants to determine the best privatizer. The steps below will eventually be automated, when users contribute tools, privatizers, and measurements to a common repository.

1. **Determine attack coverage**: First, it is necessary to specify (a) private properties (i.e., secondary uses) from shared data and (b) select adversary tools that will carry the attack (e.g., state-of-the-art (ML) technique to extract mood information from voice).
2. **Defining the privatizers**: Next, the users specifies the set of privatizers to be evaluated; each privatizer is applied to D_{raw} to generate each D_{priv}.
3. **Assessing energy**: A user defines an energy budget and measures the energy consumption of each privatizer on its IoT device. This involves examining the device-specific sensing-cycle and measuring privatizer's energy use for each component presented in Table 2.
4. **Evaluating utility and privacy**: The utility and privacy loss for each privatizer needs to be evaluated on each D_{raw} and D_{priv} with the appropriate task performance metric (e.g., Accuracy, Recall, or F1 score).
5. **Choosing the privatizer through the UPE model**: Determine the weight values for the UPE model objective α (Utility), ω (Privacy), and σ (Energy), and compute the optimal privatizer using the method described in this section, essentially computing Equation 8. Users may want to perform a weight sensitivity analysis to better understand the privatizer tradeoffs.

3.3 Implementation & Evaluation

We evaluate our UPE framework using two case studies — audio and image. For both these studies, our evaluation setup consists of the following. As shown in Figure 8, the experimental setup consists of a Raspberry PI 3B (RPi3) as our IoT device, a Pi Camera, a

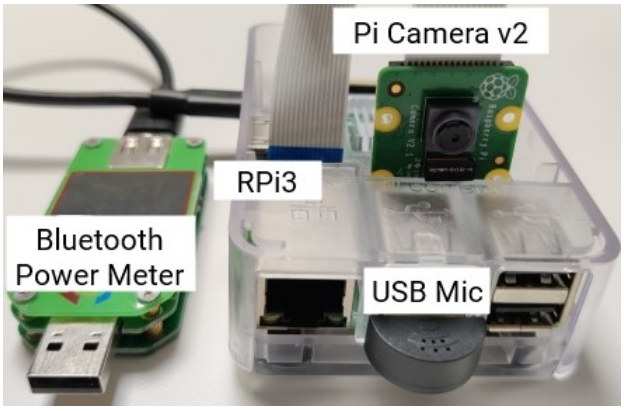


Figure 8: Experimental setup

USB mic, and a power meter to measure the current. To isolate the energy consumption of the privatizers and other components, we developed a Privatizer Evaluation System (PES) and a Sensing-Cycle Simulator (SCS) in Python. PES pipelines the execution of privatizers and NN models in the RPi3. This allows us to execute various utility-privacy scenarios and collect performance metrics. SCS activates sensors and synchronizes measurement to capture the energy footprint of multiple components, including the privatizer execution.

3.3.1 Hardware

The tasks used as utility and privacy are executed and evaluated in a PC with an AMD 3900x CPU and 2070 NVIDIA GPU (the "Cloud"). The "sensor" energy consumption was measured on a Raspberry Pi 3B (henceforth "RPi3") as the IoT device, which has an ARM cortex-53 CPU with four cores and 1 GB of RAM. As discussed in Section 2.1.4 energy consumption can affect the visibility of IoT deployments. We used RPi3 to facilitate the implementation of privatizers, given the broad availability of libraries and access to plug-and-play sensors (advantages of running Linux).

We also used a Pi Camera V2, a USB mic, and a Bluetooth power meter connected between the RPi3 and the power source. This setup was used for both the image and audio and for measuring the energy consumption of privatizers. The impact on the utility and privacy of privatizers is measured based on the performance of pre-trained DNN models on

publicly available data sets for the image and audio tasks.

Implementation. We implemented a Privatizer Evaluation System (PES) and a Sensing-Cycle Simulator (SCS) in Python. PES facilitates the evaluation of privatizers and the execution of Utility and Privacy models. We implemented the Utility and Privacy models as inferences with Neural Networks. On the other hand, SCS assists designers in measuring energy consumption by simulating customizable sensing-cycle stages.

PES allows loading large dataset files, data pre-processing, privatizer execution, and NN inference. PES also includes an asynchronous checkpointing mechanism that persists intermediary results (in case of faults). PES controls the data flow between components as follows:

1. Loads user-defined NNs (TensorFlow or PyTorch)
2. Iterates over each input file and applies a privatizer by calling an interface method that implements each candidate privatizer
3. Applies utility and privacy task models to the privatized data
4. Collects outputs and computes performance metrics
5. PES also runs once on raw data to collect the baseline performance.

SCS implements interface methods for each stage of the sensing cycle. A sensing interface simulates the sensor activation, which is called for a given interval to collect energy consumption measurements. A privatizing interface allows users to specify each privatizer and the path for an input file. SCS will apply each privatizer on each input and collect its average execution time. SCS allows for synchronizing the measurements with the external USB power meter. Finally, an uploading method will repeatedly upload the file to a user-defined URL to measure the network speed. Each stage and privatizer execution is repeated for 10 minutes.

3.4 Image Case Study

In this case study, the *utility* is a people counting task using images, while the *privacy* task is protecting identity (i.e., face recognition); we are concerned with energy consumption.

3.4.1 Performance Metrics

Utility. The performance of people counting was measured with a F1 score defined as:

$$F_1 = TP/(TP + 0.5 \cdot (FP + FN)) \tag{9}$$

where the true positives (TP) are defined as matching the number of people in the image, while extra detections (not in the original image) are a false positives (FP) and missed detections are false negatives (FN). The F1 with D_{raw} and D_{priv} for each privatizer s is then used calculate $U(s)$ (Eq. 2).

Privacy. We also used F1 score (Eq. 9) to calculate privacy loss $P(s)$ (Eq. 4). Each face match within a threshold is classified as the same person [126]. Since we compared original images against their privatized/modified counterparts, any face match is considered a TP, any extra face detections not in the original image are FPs, and missing a face is a FN.

Energy is measured in joules as defined in Section 3.2.3.

3.4.2 Dataset

We use the COCO dataset to evaluate privatizers' UPE tasks [3]. This dataset has $5K$ images with objects (e.g., cars, people, and animals) and their respective ground truth labels. However, since the COCO dataset does not have ground truth labels for faces, we use a face detection algorithm [149] and the people bounding boxes ground truth from the COCO dataset to create the ground truth for face labels (i.e., identities).

Table 3: Privatizers evaluated

Privatizer	**Blur kernel; area covered**
Avg Blur	Average kernel; global blurring
Gauss Blur	Gaussian kernel; global blurring
Med Blur	Median kernel; global blurring
BiLat Blur	Bilateral-filter kernel; global blurring
Face Blur	Average kernel; face

3.4.3 Privatizers

We analyzed five blurring techniques as image privatizers. Blur filters are often used as an obfuscation method since they can reduce image detail by attenuating pixel values [146, 141, 124]. Previous studies showed that Gaussian blurring had poor performance in hiding information from NNs, but they did not explore different blurring kernels nor blurring intensities [97, 66]. We expand these studies by including different blur kernels and intensities while also assessing the energy costs.

Blurring is a convolution technique that filters pixels according to a kernel of size $k \times k$ (henceforth denoted k^2). The intensity of blur effects can be changed by increasing the kernel size or repeating the same convolution multiple times (i.e., number of passes). We use the notation (k^2, p) as the blurring hyperparameters, where p is the number of passes. We evaluate the following blur functions:

- *Average Blur* is a linear low-pass filter in which each pixel in the output image is equal to the average of the kernel pixels from the input image. It is typically very fast to apply.
- *Gaussian Blur* is similar to convolving an image with a Gaussian function. It is also a low-pass filter similar to the averaging blur, but it preserves object edges better.
- *Median Blur* is a non-linear filter that replaces the central element of kernels by the median value of all pixel values in the kernel box.

- *Bilateral Filter* is a non-linear filter that tends to preserve edges. The replacement of each pixel is weighted on the euclidean distance of its neighbors and can include radiometric properties (e.g., color intensity).
- *Face blur* relies on a face detection algorithm [149]; it applies average blur only within the facial region.

The secondary usage in this case study (i.e., privacy attack) is a facial recognition task, which relies on facial features.

The intensity of blur effects can be tweaked by increasing the kernel size or repeating the same kernel convolution multiple times (i.e., number of passes). As images lose texture and edge details, utility and privacy tasks will miss-classify objects and faces.

However, our goal is not to find the best privatizer but only to showcase the selection process of an energy-aware privatizer with the model proposed in Section 3.2.4.

While a DNN could be trained on blurred images for better performance, some loss is expected since blurring is a lossy transformation. Models trained on blurred data will have less information to learn since the data becomes more homogeneous as it is blurred. Also, although blurring effects are hard to reverse completely, much information can be recovered under low levels of blur [127]. However, we did not re-train the attacker DNN on blurred images in this evaluation.

3.4.4 Methodology

The people counting task (Utility) uses the "faster RCNN inception v2", a popular model, pre-trained on the COCO training dataset containing 118,000 images [54]. The face recognition task (Privacy) uses the Facenet model, pre-trained on the popular VGGFace2 dataset [126]. We used opencv-contrib [107] to implement image transformations.

To measure energy, each component repeatedly performed the same task (e.g., taking photos, applying blur, or uploading a file) for 10 minutes. We also measure the instantaneous current with a rate of 2Hz, generating 1200 samples for each component. The energy consumed by privatizers and the network was computed with the COCO dataset's average image size (163KB).

Table 4: Energy consumption for RPi3 sensing-cycle.

Energy Component	**Curr (A)**	**Time (s)**	**En (J)**
Standby/base (da)	0.308	-	-
Camera (sa)[a]	0.089	0.0504	0.013
Network Active (na)[a]	0.070	0.086	0.038
Network Idle (ni)	0.020	-	-
Avg Blur	0.460	0.1365	0.324
Gaussian Blur	0.760	0.049	0.190
Median Blur	0.462	1.0911	2.588
BiLat Blur	0.742	0.401	1.537
Face Blur	0.791	1.274	5.185

[a]Energy necessary to capture or upload one single frame at 640x480.

3.4.5 Energy Consumption Analysis

We begin by analyzing the energy consumption of different privatizers' energy components. Table 4 summarizes the overall current, execution time, and energy for blurring and uploading an image from a camera. We note that while some privatizers may require a high current, their execution time may also significantly vary, thus changing the energy consumption. For example, the current of BiLat Blur privatizer is 0.74A, whereas the current of Median Blur privatizer is 0.46A. However, median blur takes longer to execute, increasing its overall energy footprint. On further investigation, we notice that BiLat Blur uses all the CPU cores, which reduces the overall execution time but increases the overall current. Thus, a key takeaway is that designing energy-efficient privatizers will require attention to the execution time and power-performance tradeoff.

We also analyze how various hyperparameter configurations of privatizers affect energy consumption. In this experiment, we vary the kernel size of the blurring algorithm and measure its energy consumption. To our surprise, the hyperparameters not only affect the overall energy, but as we vary the hyperparameters, the most energy-efficient privatizer for a configuration can become the least energy efficient for a different configuration. As shown in Figure 9, the Gaussian Blur is the most energy-efficient when the kernel size is less than

4. However, as we increase the kernel size, it becomes least energy-efficient. We also see similar behavior in Bilateral Blur. Thus, it is important to evaluate the energy consumption for different hyperparameters to identify energy-efficient privatizers. This also helps define the scope of hyperparameters to continue the UPE evaluation.

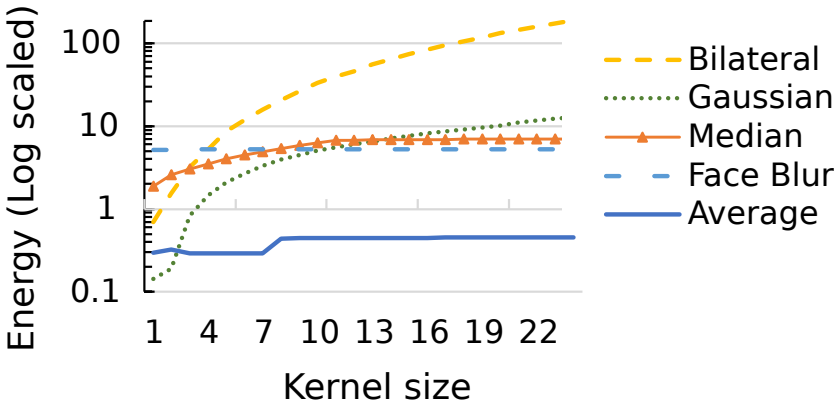
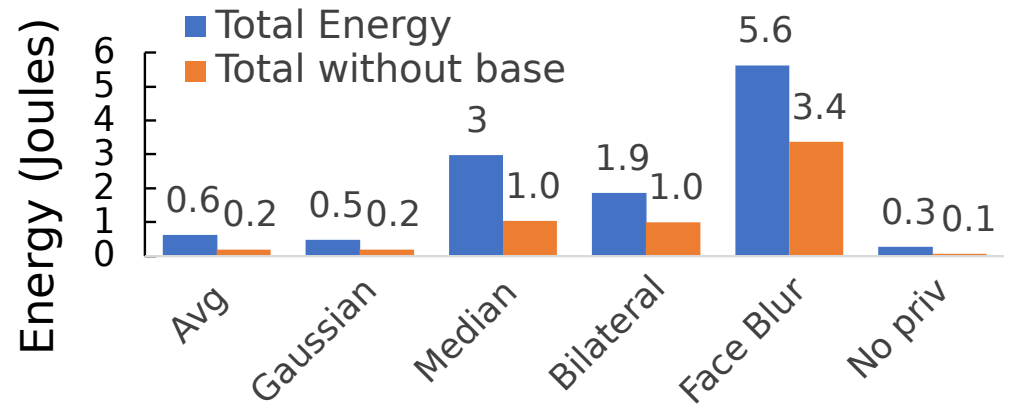


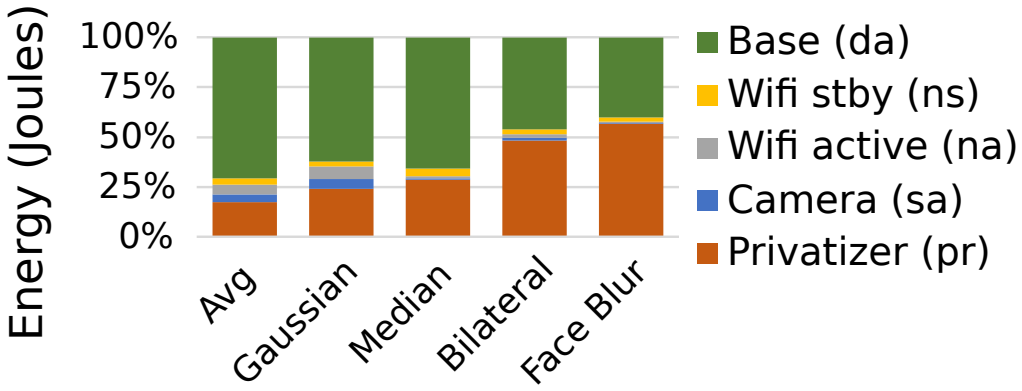
Figure 9: Energy consumption for different video blur kernels

Figure 10 shows the impact of each device component on the total energy consumption and highlights the role of including the base energy. As mentioned earlier, the duration that components remain active will affect the overall energy consumption. In particular, Figure 10(a) shows that the device base energy has a varying impact on the total energy consumption of each privatizer. For example, the base energy tripled the total median blur energy while it only nearly doubled the Bilateral Blur energy. Further, the relative energy used by each component and privatizer is presented in Figure 10(b). The energy used by the network (both standby and active Wifi) represents less than 7% for all privatizers, and the base energy is dominant for Avg, Gaussian, and Median. In comparison, the energy used by Face and Median blurs represents 57% and 29% of the total, respectively. This shows that for the RPi3 device, the base and privatizer energy consumption dominate energy consumption.

Measuring the base energy RPi3 energy consumption allows the comparison between platforms. For example, energy-efficient platforms (e.g., ESP32) have lower base energy consumption due to simpler circuity, processing unit, and fewer features than an RPi3. Also, other devices will have different consumption profiles for each sensing-cycle stage due to processing and network bandwidths.



(a) Total energy



(b) Relative energy use by sensing-cycle component

Figure 10: Privatizers' total and relative energy consumption for video privatizers (5 blur passes and 5^2 kernel size)

Summary. These experiments show that some privatizers' hyperparameters can be discarded from further analyses if the privatizer already consumes too much energy. Privatizers with linear energy consumption scaling (Average blur) can be more flexible when needing to vary the intensity of the effect. Also, the device base energy *da* represents a considerable portion of the total while the sensor itself (camera) and network are almost negligible. Moreover, we show that the main drivers of energy consumption are the sensing-cycle duration and not necessarily the most CPU-efficient privatizer.

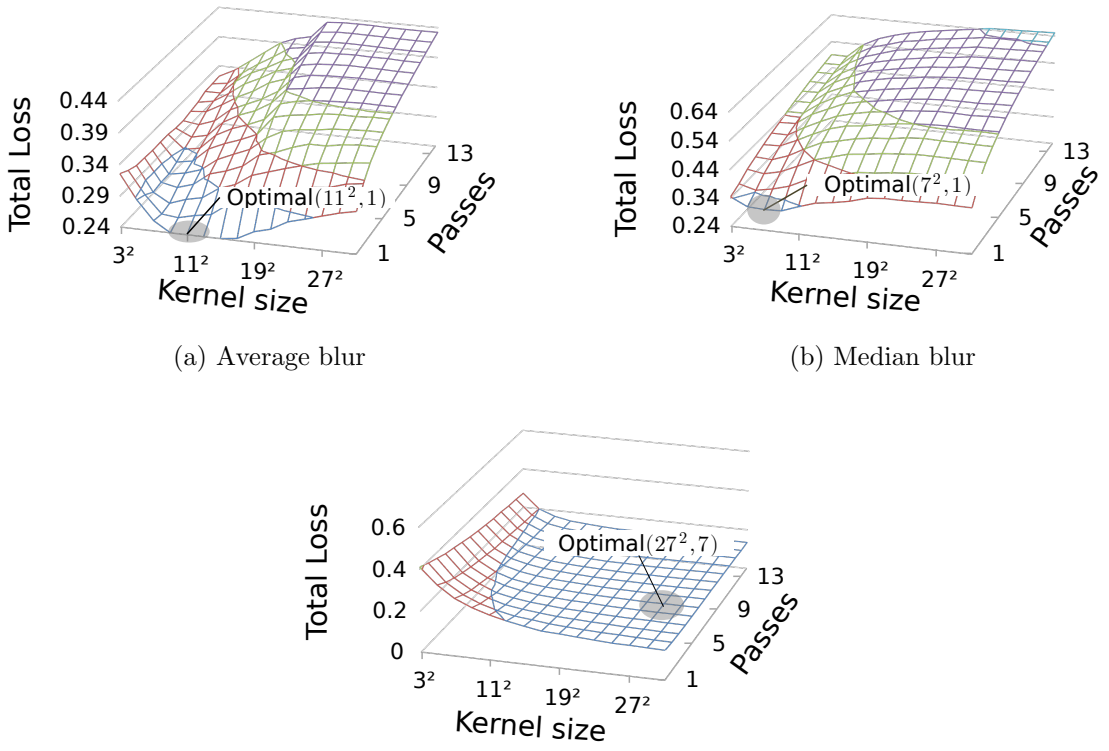
Moreover, the face blur illustrates the energy consumption tradeoff for isolating the sensitive portion of the data (face) and then applying the effect in a smaller region. Finally, we also emphasize the importance of including the energy consumption of devices base energy *da* and not only of the privatizer execution (i.e., privatizer CPU usage). For example, the format and size of sensed data can be modified to accommodate a less intense privatizer and save energy for the sensor and processing time (e.g., less intense blur for lower resolution images).

Also, prohibitively costly privatizers can be discarded from further analysis (e.g., bilateral blur with kernels above 11^2 in Figure 9), reducing the range of hyperparameters and privatizers candidates.

3.4.6 Privatizer Selection

We now analyze the effect of different hyperparameter configurations on utility, privacy, and energy. To do so, we assign equal weights to utility, privacy and energy, i.e., $\alpha = \sigma = \omega = 0.33$. Further, we vary the kernel size and apply the blur effect across multiple passes. Note that applying the same blur effect across multiple passes affects utility, privacy, and energy. Thus, we can analyze the tradeoff surface across different hyperparameter configurations to identify the best privatizer that meets application constraints.

Figure 11 show the total loss across different hyperparameter configurations for three different privatizers. We observe that it is unclear apriori which hyperparameter configuration results in the best UPE tradeoff. For example, the average blur privatizer in Figure 11(a) indicates that the optimal privatizer that achieves the lowest loss has a hyperparameter con-



(a) Average blur

(b) Median blur

(c) Face blur

Figure 11: Sensitivity analysis of Average, Median, and Face blur with $\alpha = \sigma = \omega = 0.33$

figuration with kernel size 11 with a single pass. However, as shown in Figure 11(c), face blur achieves the lowest loss with a hyperparameter configuration of kernel size 27 and 7 passes.

Furthermore, the total loss as a function of its hyperparameter configuration is not monotonic in nature. We observe that average and median blur losses decreases and increases with larger kernel sizes. In particular, the total loss in average blur drops from 0.32 to 0.23 when we vary the kernel size from 3 to 11. But, it increases to 0.43 when we increase kernel size to 27. This indicates that the tradeoff surface may be irregular, and we need to perform a search to identify the best privatizer within its hyperparameter space. Note that performing this grid search is a one-time operation and thus not very expensive to compute. Moreover, as seen in Figure 11, trends may emerge, and we can use them to limit the grid search space.

Table 5: Image UPE fit function sensitivity analysis

	Avg Blur		**Median Blur**		**Face Blur**	
	(k,p)	loss	(k,p)	loss	(k,p)	loss
U[1]	(3,1)	0.03	(3,1)	0.05	**(3,1)**	**0**
P[1]	**(31,15)**	**0**	**(31,13)**	**0**	(31,13)	0.011
E[1]	**(7,1)**	**0**	(3,1)	0.012	(9,1)	0.33
UP[2]	(11,1)	0.35	(9,1)	0.38	**(25, 8)**	**0.034**
PE[2]	**(31,1)**	**0.04**	(31,1)	0.18	(27,7)	0.17
UE[2]	**(3,1)**	**0.015**	(3,1)	0.08	(3,1)	0.17
UPE[3]	(11,1)	0.23	(7,1)	0.31	**(27,7)**	**0.13**
2UPE[4]	(7,1)	0.24	(3,1)	0.27	**(25,6)**	**0.11**
U2PE[4]	(21,1)	0.24	(31,1)	0.31	**(31,13)**	**0.10**
UP2E[4]	**(11,1)**	**0.17**	(9,1)	0.28	(31,4)	0.18

[1]Single objective with weight of 1 [2]Two Objectives with weight of 0.5

[3]Equal weights (0.333) [4]One objective with 2x more weight (e.g., 0.5, 0.25, 0.25)

Model sensitivity analysis. Finally, we analyze how different UPE objectives can affect our privatizer selection process. To do so, we assign different values to the weight parameters in our UPE objective. If the objective is utility and privacy, we can assign $\alpha = \omega = 0.5$ and $\sigma = 0$. Similarly, if the objective is to identify an energy-efficient privatizer, we can assign $\sigma = 0.5$ and set $\alpha = \omega = 0.25$. We also experimented with other privatizers (e.g., bilateral and Gaussian) but observed that average, median, and face blur achieved the lowest loss. Thus, we report the loss for only average, median, and face blur privatizers.

Table 5 shows the overall loss for different hyperparameter configurations of privatizers and UPE objectives. As shown in the table, depending on the weight chosen for each objective, the choice of a privatizer may change. For example, if the objective is to preserve utility and privacy, our result indicates that we should use face blur. However, if the objective is to preserve privacy (or utility) and energy, the face blur privatizer is not ideal and should choose an average blur privatizer over face blur. Similarly, if we give equal importance to UPE, we observe that the face blur privatizer achieves the lowest loss. We note that

average blur still achieves the lowest loss if we change the objective to identifying the most energy-efficient privatizer.

3.5 Audio case study

In this case study, the *utility* is speech-to-text (S2T) translation, the *privacy* task is voice recognition.

3.5.1 Performance metrics

Utility. We measure utility using Word Error Rate (WER), a common metric for S2T: lower WER means better performance. WER is computed as $WER = (s + i + d)/N$, where s is the number of substituted words, i is the number of new inserted words, d is the number of missed words, and N is the ground truth for the number of words.

Privacy. We use the cosine distance between two voice embeddings to identify an individual [121]:

$$P(D_{raw}, D_{priv}) = 1 - \frac{f(D_{raw}) \cdot f(D_{priv})}{\|f(D_{raw})\| \cdot \|f(D_{priv})\|} \tag{10}$$

where f is a DNN trained to extract people's unique voice features as embeddings, note that the cosine distance ranges from 0 to 1, where 0 means the voice embeddings belong to the same person. Moreover, if the distance is larger than a threshold, it means a different person.

Energy is defined in Section 3.2.3.

3.5.2 Dataset

We used the TIMIT corpora, which contains 630 speakers of eight major dialects of American English and contains the ground truth as transcriptions. The evaluation test set contained 168 speakers [5].

3.5.3 Privatizers

We evaluated five audio effects as privatizers, namely:

- *Pitch shift* lowers or raises the pitch of audio signals at preset intervals (e.g., semitones). We explored pitch values ranging from -440 to 440 in steps of 40.
- *Tremolo* is a modulation method that adds variation in depth and modulation frequency to the amplitude of an audio signal. It can create percussive stuttering effects. As a privatizer, it is expected to deform the amplitude wave shape and disrupt the voice recognition model. We varied depth (0, 10, ..., 100) with a 500 Hz tremolo effect.
- *Reverberation* effects add resonance to audio signals and are known to disrupt automatic speech and speaker recognition [84]. We used values ranging from 100 to 0 for reverberance in steps of 10.
- *Mixed* mode includes tremolo, reverberation, and pitch shift (all together) ranging in intensity as described above.
- *Audio Blur (AB).* We developed a new technique inspired by the image blurring presented in Section 3.4 as a privatizer. AB applies a sliding window over the audio amplitude vector, substituting the central value with the average once. The intensity of AB can be controlled by increasing the size of the window (similar to image blur kernels). We evaluated the following window sizes: $(3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23)$.

The first three effects described above are commonly used in audio applications [27]. Also, since the voice recognition model uses a Convolutional NN, the Audio Blur effect will attenuate high values in the audio signal and, we hypothesize, hinder the models' performance. Further, we report the results by normalizing the hyperparameters of the audio effects within a 1 to 10 scale. For example, an intensity of 5 represents a positive (negative) pitch shift of 220 (-220) semitones or a 50 depth tremolo; these values are half the range of the effects.

3.5.4 Methodology

The S2T (utility) was implemented with Mozilla's DeepSpeech, a Recurrent NN model trained on several datasets consisting of 5,000 hours of speech [58]. The voice recognition task (Privacy) used the SincNet model, a CNN designed to extract voice embeddings, trained with the TIMIT and Librispeech corpora [121]. We compared the voice embeddings for the same audio recording against its privatized version. We used the pysox [118] library to manipulated the audio data.

Also, we used a USB mic as the sensor, and the energy used by audio privatizers was computed over 10 minutes for each audio effect and hyperparameter pair. The average audio duration (2.43s) and file size (94KB) of the TIMIT dataset was used to compute the energy consumption of the audio sensor and the time to upload the audio files for both the *sensing* and *uploading* stages.

3.5.5 Energy Consumption Analysis

Now we analyze the energy consumption of different energy components for the audio use case. In addition to the device components already measured in Table 4, we add the audio sensor (mic), measured at 0.04A. Figure 12 shows the total and relative energy consumption for each privatizer with medium (5) intensity. Privatizer's processing (bottom portion of the bars in orange) had a small relative impact on the total consumption while mic, network, and base consumed most of the energy; actually, base consumed more than 80% (top portion of the bars, in green). Also, the most energy-efficient privatizer is Audio Blur, consuming 3% to 10% less energy than others.

Also, we note that the different hyperparameters for the audio effects have a small impact on energy consumption compared to the image case study privatizes. The audio privatizers' execution time ranged from 0.1 to 100 milliseconds, and their implementation is single-threaded, adding only 0.15A.

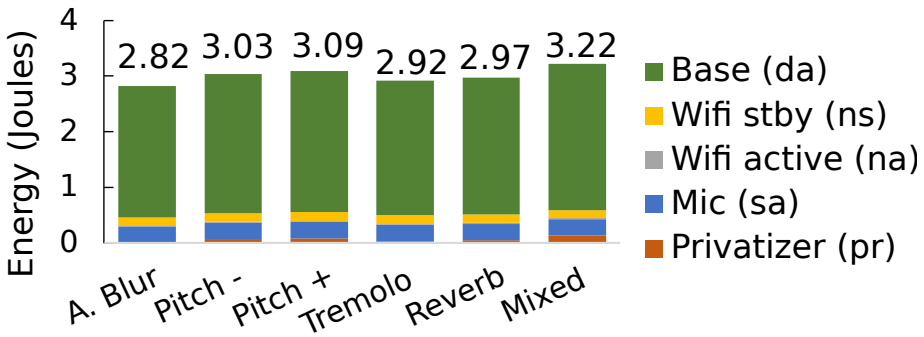


Figure 12: Audio privatizers energy consumption.

3.5.6 Audio Privatizers' Selection

Similar to images, we explored each privatizer with different intensity levels by tweaking their hyperparameters. Figure 13 presents the results for the audio UPE model. Our proposed Audio Blur reached the lowest loss by significantly degrading the attack model while keeping the DeepSpeech model performance almost intact. All the other effects equally affected both privacy and utility. Reverberation was the worst, with minimal impact on privacy, high utility loss, and high energy consumption. The mixed effect has the best privacy loss.

Model sensitivity analysis. Table 6 presents the optimal (in bold) intensity levels i and *total loss* for each audio privatizer with different α, ω, σ weight values for the fit function. Lower values are better since we want to minimize them. Audio Blur is the optimal solution since it has the lowest loss for most weight combinations. However, a level 1 intensity of positive pitch has the lowest utility loss, while a level 10 intensity of the mixed effect has the lowest privacy loss. The optimal intensity for the audio blur is 8 for the UPE with equal weights. Pitch shift and tremolo energy consumption did not vary since their algorithms have a constant complexity for their intensity hyperparameters. Pitch shift, tremolo, reverberation, and mixed mostly degraded the S2T performance rather than the Speaker recognition, implying that these privatizers mainly change S2T features. Finally, note that Audio Blur is the optimal privatizer, having the lowest loss for most UPE weight

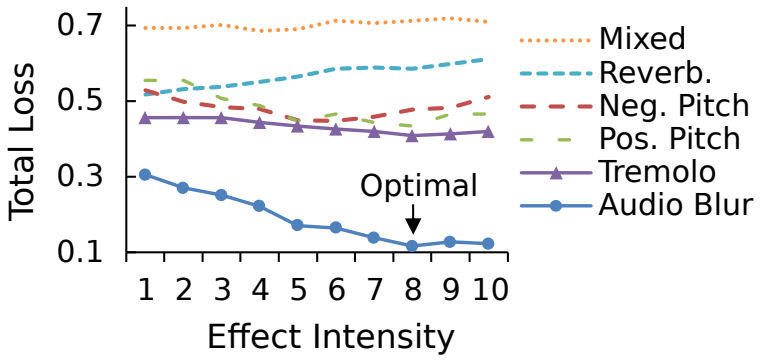


Figure 13: Effect of privatizer's intensity.

selections.

3.6 Related Work

Table 7 presents the coverage, application focus, and privatizers techniques considered in related works that modeled UPE, UP, and UE objectives. We note that most solutions are application-specific with fixed data types, privatizers, or UPE metrics [43]. In [94, 43], researchers attempt to capture a generic interpretation of privacy loss with information-theoretic metrics (e.g., mutual information). However, these approaches do not always easily translate to the performance of specific SU attacks. In [94, 15], a privatizer is developed with autoencoders to extract only the useful content while removing other information by minimizing the mutual information between D_{raw} and D_{priv}; however, the energy cost is not explored. In [43] and [30], the UP tradeoff is studied in the context of data frequency collection of smart meters and its relation to SU attacks. The work in [43] assess how the collection frequency relates to different services, while the work in [30] explores the use of batteries to hide the load of devices that can leak occupancy information.

Utility-energy tradeoffs have been recently studied in the use of NNs on embedded devices in [76] and [142], where optimizations rely on the efficient management of hardware resources,

Table 6: Audio UPE fit function sensitivity analysis

	Audio Blur	**Pitch-**	**Pitch+**	**Tremolo**	**Reverb**	**Mixed**
	(i, loss)	(i, loss)	(i, loss)	(i,loss)	(i, loss)	(i, loss)
U[1]	(1, 0.02)	(1, 0.01)	**(1,0)**	(2, 0.01)	(1, 0.13)	(1, 0.12)
P[1]	(9, 0.04)	(10, 0.16)	(10, 0.07)	(10, 0.69)	(10, 0.81)	**(10,0)**
E[1]	**(1,0)**	(9, 0.47)	(3, 0.56)	(-, 0.21)	(1, 0.31)	(1, 0.85)
UP[2]	**(8, 0.09)**	(5, 0.35)	(7, 0.28)	(7, 0.43)	(1, 0.55)	(10, 0.5)
PE[2]	**(9, 0.03)**	(10, 0.32)	(10, 0.34)	(10, 0.45)	(10, 0.57)	(10, 0.5)
UE[2]	**(1, 0.01)**	(2, 0.27)	(1, 0.3)	(2, 0.11)	(1, 0.22)	(1, 0.49)
UPE[3]	**(8, 0.07)**	(6, 0.39)	(8, 0.38)	(8, 0.35)	(1, 0.46)	(4, 0.63)
2UPE[4]	**(8, 0.08)**	(5, 0.34)	(5, 0.32)	(7, 0.29)	(1, 0.38)	(1, 0.52)
U2PE[4]	**(8, 0.06)**	(8, 0.38)	(10, 0.33)	(10, 0.45)	(1, 0.59)	(10, 0.5)
UP2E[4]	**(8, 0.05)**	(6, 0.41)	(8, 0.42)	(8, 0.32)	(1, 0.42)	(1, 0.69)

[1]Single objective with weight of 1 [2]Two Objectives with weight of 0.5 [3]Equal weights (0.333)

[4]One objective with two times more weight (e.g., 0.5, 0.25, 0.25)

allowing the processor to operate at lower voltages coupled with NN compression. In [142], researchers show how FLOPS are not a good metric for the execution of NNs on embedded devices, recommending execution time instead. Moreover, the cost of IoT devices' standby energy is explored in the context of user experience and device responsiveness. Finally, in [15], energy is assessed by measuring the CPU time of autoencoders. In contrast, we show that CPU time can be insufficient to characterize privatizers' energy consumption in IoT devices since other components also play a role.

Rate-distortion-equivocation from information theory has been extended to solve the UP problem [41], presenting a framework focused on structured data with known distributions that determine an analytical model's optimal UP tradeoff. This framework can model generic data sources and create privacy-utility tradeoff metrics bounded by user-defined constraints. Still, their model does not generalize to non-independent and identically distributed random data sources with unknown distributions and unstructured properties (e.g., Web searches). Also, they use Shannon's entropy as a fixed metric to determine privacy leakage, which does

Table 7: Related work comparison

Work	**Objectives**	**Aplication**	**Privatizer**
Ours	**UPE**	**any**	**any**
[15]	UPE	audio	AE
[43]	UP	any	noise
[124]	UP	videos	obfuscation
[94]	UP	time series	AE
[30]	UP	smart-meters	randomization
[142, 80, 76]	UE	neural nets	-

not offer accessible information on the success rate of specific attacks. In contrast, our model is data agnostic and we could incorporate their utility and privacy metrics.

We note that simultaneously addressing energy, privacy, and utility has not been thoroughly explored since energy typically takes a second seat (lower priority) when considering privacy mechanisms. Our work explores the UPE problem and proposes a framework to guide the selection of the best privatizer for any application or sensed data type.

3.7 Discussion

It is important to note that while UPE framework can guide users to find the best privatizer along the UPE tradeoffs, the selection of the weights can change the selection of the best privatizer, which can play a crucial role in the usability of the IoT system as discussed in Section 2.1.4.

For mobile battery-operated systems (e.g., Section 2.1.4), it is also essential to consider the usability impact of high energy-consuming privatizers since the energy score, as defined in our model, may hide the usability consequences, such as replacing the battery of a deployed system more often. Also, in systems that harvest energy for continuous operation (e.g., solar or aeolic), higher-energy privatizers may be prohibitive, and the user has to define a minimum energy budget as a constraint along with utility and privacy minimum requirements.

3.8 Conclusion

This Chapter's work was targeted at answering $RQ1$ through creation of foundational concepts. We designed, developed, and evaluated a framework for characterizing privatizers' Utility, Privacy, and Energy (UPE) tradeoffs for IoT applications. We evaluated our model with two case studies: image and audio data. Results and analysis show the following insights: (i) UPE tradeoff space is not trivial, requiring experimentation to find efficient privatizers; (ii) our framework shows how to select privatizers that fit user-defined priorities by minimizing a fit function that optimizes utility, privacy, and energy objectives; (iii) privatizers that target only the sensitive features can potentially minimize the impact in utility with an extra cost in energy for finding the sensitive features (e.g., blurring only facial region with face detection) as shown in the image case-study.

4.0 Developing Utility-Aware Privacy-Preserving Tools in Federated Learning: An Examination of IoT Smart Meters

4.1 Introduction

In Chapter 3, we evaluate the protection of privatizers against inferences based on pre-trained models. In this Chapter, we study the use of federated learning and differential privacy to protect users' data when used to train neural network models. We focus on smart meter data as a common type of IoT sensor.

Utility companies are rapidly deploying smart meters capable of measuring and transmitting fine-grained aggregate energy consumption. This data has many use-cases, including non-intrusive load monitoring (NILM) that help users estimate individual appliances' power consumption without the need for costly appliance-level instrumentation. While NILM can help in power-efficiency measures [74, 25, 59] (e.g., find inefficient appliances), fine-grained power data raises privacy concerns as they can reveal private information (e.g., number of people or sleeping habits) [99, 100].

As introduced in Section 2.1.6, Federated learning (FL) techniques have emerged as a solution that can train models across multiple users while keeping the data local to these users. This reduces the privacy risks resulting from sharing user data to a centralized server, mitigating regulatory and security concerns. However, the privacy guarantees provided by federated learning are limited. Prior studies show that federated learning is vulnerable to privacy attacks, including membership inference and attribute inference attacks [71]. A membership inference attack happens when the adversary wants to determine if a particular data point was used in training a machine learning model, potentially revealing sensitive information linked to the model. An attribute inference attack infers sensitive attributes of an individual based on other available information or outputs from a machine learning model, even if the data is anonymized. This may reveal private information about the clients involved in the training process [71, 144]. Prior studies have proposed differentially private federated learning (DPFL) to provide clients with stronger privacy guarantees [8]. This

approach adds calibrated noise to the data using differential privacy techniques to prevent information leakage [46]. That is, each client perturbs the information before sending it to the server, thereby preventing information leakage.

This Chapter studies the use of DNN NILM models with differentially private federated learning. We note that for systems to be truly private, they must avoid sharing local client data and adopt privacy mechanisms (e.g., DP) that prevent privacy leakage information in a principled way. Users should be able to track how much information is sent to the remote service, thereby providing more control over their data. In a DP framework, this is done by examining the privacy loss dictated by the ϵ and δ parameters in the (ϵ, δ)-DP framework. A privacy accountant tracks and stops the training once the privacy limit reaches a threshold [8].

Towards this end, we develop a differentially private federated learning (DPFL) framework for training NILM models. To answer $RQ2$ we design a framework that adds a new capability for evaluating NILM models in the DPFL setting. The key goal is to provide a privacy-preserving distributed framework for training and enable practitioners to study various NILM models and their effectiveness in mitigating privacy attacks within a DP framework. We build upon the NILM toolkit [26] that supports multiple NILM algorithms and datasets.

Moreover, we employ the TensorFlow Framework (TFF) [4] for training and managing the simulation of the federated learning client-server interactions. In particular, TFF manages the client-server communication of model updates used to create the federated model. Doing so allows our framework to be extensible, where we can plugin existing models and evaluate their performance on various attack metrics and datasets. Our framework integrates NILM toolkit with TFF to facilitate the utility-privacy tradeoff evaluation of training DNN models in a distributed fashion.

4.2 Background

NILM. NILM techniques aim to recover each appliance's power consumption, given only

the aggregate power measurements. Formally, let $P = (P_1, \ldots, P_t)$ represent the aggregate power consumption readings from a smart meter for a given time interval, where P_t denotes power measured at time t. Further, let us assume there are N appliances, and for each appliance, i its instantaneous power consumption at time t is denoted by A_{it} where the total power consumption up to time t is the set $A_i = (A_{i1}, \ldots, A_{it})$. Then, we know that the aggregate power $P_t = \sum_i^N A_{it} + \Delta_t$, where Δ_t is the noise term. NILM identifies the contribution of each appliance A_i towards the aggregate power given P while also allowing to infer on/off events for each appliance [26].

4.2.1 Differentially Private Federated Learning

Federated learning is a decentralized training architecture where a trusted curator (e.g., utility companies) trains a shared model by communicating model parameters with clients (e.g., smart meters). It ensures that the raw data remains local to clients, minimizing attack vectors in transmission and remote processing. As proposed in [95], each client downloads the latest model parameters from the trusted curator and trains a local model. After training, each client then uploads the trained parameters back to the curator. Then, the curator aggregates the updated parameters to create a central model. However, even though no raw data is shared, federated models are still susceptible to privacy attacks, such as membership inference attacks (MIA) [70].

Prior work has proposed training models within the differential privacy frameworks to address the privacy concerns in federated learning. Differential privacy reduces the influence of any specific user's data by amortizing its effect on the aggregated trained model. To do so, it introduces calibrated noise (e.g., using the Gaussian mechanism) during the training process to control the influence of the user's data over the model. This noise is controlled by the ϵ parameter and calibrated to the function dataset sensitivity S_f, defined as the maximum absolute distance $|f(d) - f(d')|$, where d and d' are adjacent inputs (i.e., datasets that differ on just one record). Further, for a given ϵ, a privacy accountant tracks the current *epsilon* value and stops the training once a maximum *epsilon* value is reached. This maximum *epsilon* value is also called the privacy budget since, and at each training round, the epsilon

value increases as the model learns more about the users' data. The designer must choose how much epsilon he will use to train the final model at the cost of increased privacy leakage. We refer the readers to [95] for more information. Recent work has also been on using Renyi differential privacy (RDP), which provides much tighter privacy guarantees [23]. TensorFlow privacy framework implements RDP, which we adopt in our framework.

4.2.2 NILM and FL Platforms

There has been much work on developing NILM techniques, federated learning, and differential privacy platforms [26, 59, 56, 34]. Our framework builds upon these foundational building blocks. For example, NILMTK toolkit [26] provides a framework to evaluate multiple NILM techniques. Similarly, Tensorflow and Torch-based platforms exist that enable FL. Although our framework's underlying techniques are not novel, our approach supports and simplifies the development of privacy-preserving NILM models. Thus, our novelty lies in the ability to extend our framework and study the privacy threats posed by private and non-private models.

4.2.3 Threat Model

The adversary in this chapter will try to de-anonymize users by trying to ascertain the involvement of specific users in the training dataset. This threat model encapsulates a potential privacy breach wherein the adversary could acquire sensitive information about users' energy usage habits and patterns, revealing personal aspects of their household activities.

The adversary in this model is assumed to be able to query the final model based on the aggregated updates from the federated learning process, and their objective is to discern whether a user's data was part of the training dataset or not. It is worth mentioning that this threat model considers the context where the utility of the service provided by the federated learning system is non-intrusive load monitoring. This means that, in addition to the privacy risks posed by the adversary, the system is also challenged to maintain its utility, accurately monitoring and providing insights on energy consumption patterns.

4.3 DP-Federated NILM Design

Figure 14 depicts our proposed architecture and consists of three layers. The key objective is to enable researchers to train privacy-preserving federated models, evaluate the performance against privacy attacks, and perform NILM tasks. These layers are built upon existing frameworks that are easily extensible to incorporate newer models, datasets, and test cases. Below, we detail each layer.

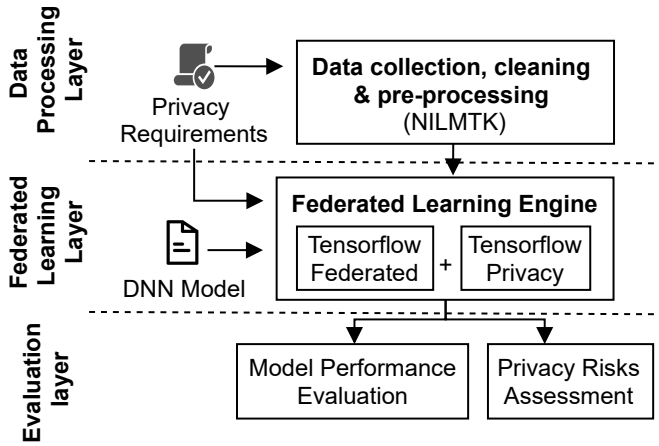


Figure 14: Privacy-Preserving Federated Framework Architecture.

Data Processing Layer: This layer leverages NILMTK to facilitate data processing and integration into the federated pipeline while exposing the data in standard formats. However, unlike NILM, we allow users to specify the number of clients participating in the training process. Our processing layer provides the capability to combine disjoint datasets and divide them into different clients in preparation for the federated learning setup. Each client thus receives a *local data* for the training period. Moreover, we can also specify the privacy requirements, controlled by the ϵ parameter. Our framework also allows customizable function calls that enable users to add pre-processing functions to either remove sensitive information or prepare the data for the model's input format.

Federated Learning Layer: This is the core layer that simulates the model training in a decentralized manner. We built abstractions over the Tensorflow Federated (TFF) frame-

work that integrates with the data processing layer [4]. We also provide an interface to integrate with various deep learning-based NILM models, facilitating developers' integration of new models while using NILMTK datasets for training. Within the DP-federated framework, model training is performed as follows. First, the model's initial state is defined and propagated to a subset of randomly selected clients, which helps amplify privacy [23]. Then, each client trains a local model and sends local updates to be aggregated in the server model. A privacy accountant can be defined to keep track of the privacy loss for each round of client subsampling, training, and server model update. This process is repeated until the privacy budget reaches a certain threshold. Since we leverage TFF, developers can customize these different steps and change various hyperparameters and optimizers (e.g., Stochastic Gradient Algorithm, Adam) to train the model.

Evaluation Layer: We extend the framework and develop an interface to evaluate the model's performance for different metrics and privacy attacks. We also provide an interface that enables developers to explore the model against privacy risk. We note that a popular metric to measure privacy leakage is to determine whether a user participated in the training to build a model [145]. Our framework includes a MIA as defined in [145], which helps quantify the benefit of privacy-preserving mechanisms such as differential privacy. The membership attack in [145] uses a threshold-based scheme to predict whether the client participated during training. Although our approach implements only a threshold-based membership attack [129], other attacks can be introduced and remain part of future work.

The proposed framework adds the capability for researchers to study the DPFL setting for different neural network models and better interpret how the *epsilon* budget relates to practical attacks. This enables the creation of models that can better protect users' privacy against membership inference attacks.

4.4 Evaluation

In this section, we describe the datasets, methodology, and results of the privacy-preserving federated learning framework.

4.4.1 Methodology

Datasets: We use existing NILM datasets to evaluate our approach, including REDD and UKDALE [78, 75]. REDD is the first publicly available dataset containing whole-house and appliance-level power consumption. We used the low-frequency version containing several appliance meters from six houses with a three-second collection rate [78]. Similarly, the UKDALE dataset comprises power measurement records of five houses collected for the whole house and appliance level meters [75]. The REDD and the UKDALE datasets comprise power measurement records from eleven houses collected for the whole house and appliance level meters recorded at different sampling rates [75]. We used NILMTK to interpolate the data every three seconds and merged UKDALE and REDD to simulate non-independent and identically distributed (non-IID) characteristics, which is common when using distributed clients' data [96]. Note that despite UKDALE and REDD houses may contain different patterns of appliance use, deep learning-based models have been shown to still perform well for NILM tasks [25].

Experimental Setup: We evaluated our framework using the Sequence-to-Point [147] (S2P) Neural Net and trained it with only FL (i.e., No Differential Privacy) and DPFL. The FL serves as a baseline for the DPFL approach and provided an upper bound on the model's accuracy.

To train our models, we used our framework to combine REDD and UKDALE datasets and split the data to simulate 1000 clients. Further, we split the dataset into training (70%) and testing (30%) datasets. Next, we train the FL and DPFL models to disaggregate the power consumption of the fridge, kettle, and microwave appliances. We set the user sampling rate to 0.01, equating to ten random users out of a thousand to participate in the training of both FL and DPFL. Moreover, we use Adam and Stochastic Gradient Optimizer optimizers for training our model at the client and server level, respectively [96]. We also set the noise multiplier to 0.3, which controls how much noise is added during the model aggregation process. Unless stated otherwise, we use $\epsilon = 12$ as the privacy budget.

We use two different metrics to evaluate model accuracy. In particular, we use precision, recall, and F1-score to evaluate the model's performance in identifying appliance's on/off

events from the aggregate energy. Also, we use mean absolute error (MAE) to evaluate the predicted appliance power. To measure the privacy leakage, we use Attacker's Advantage (AA) metric as defined in Yoem et al. in [70]. AA is calculated as the difference in True Positive Rate (TPR), and False positive Rate (FPR) of a membership inference attack (i.e., $AA = TPR - FPR$) [144]. True Positive Rate is calculated as $TPR = \frac{TP}{TP+FN}$, which gives the ratio of correctly predicting a user as a member of the training set in relation to all positive cases. In contrast, False positive Rate is $TPR = \frac{FP}{FP+TN}$, the ratio of falsely predicting the user as a member of the training model dataset.

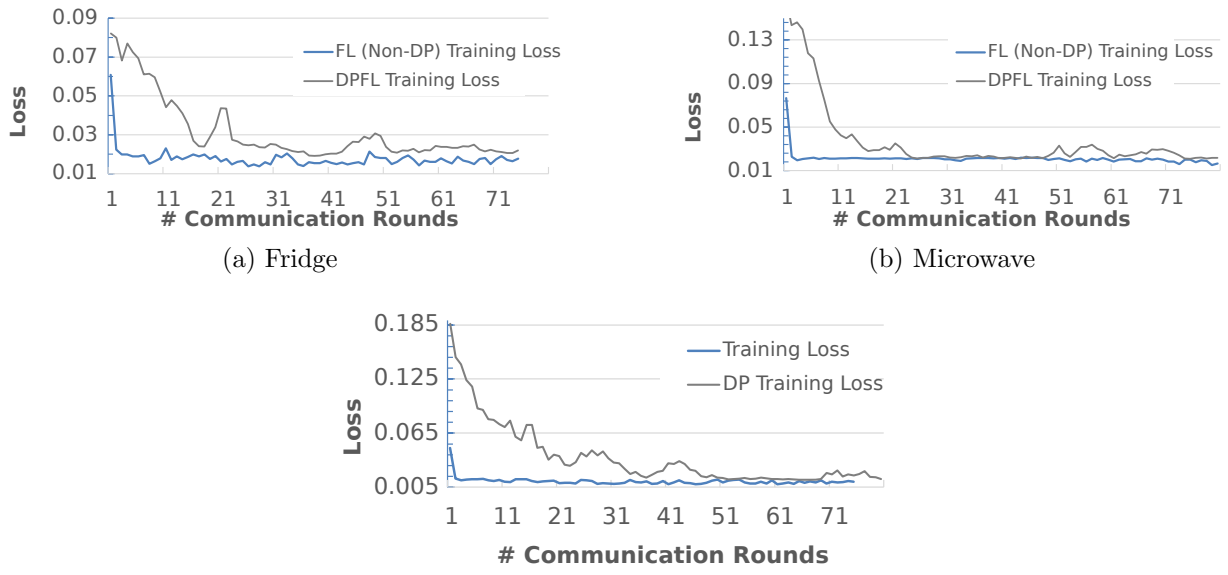
the percentage of actual positives that are accurately identified Intuitively, AA measures the improvement in a privacy attack when *members* (i.e., data in the training dataset) are included. The membership inference attack identifies members using a loss-based threshold, where the attacker uses the training loss to determine whether an input is a member or not. For more details, please refer to [70].

4.4.2 Results

Table 8: On/Off prediction metrics based on the disaggregated power signal.

	Fridge			**Microwave**			**Kettle**		
	Recall	Precision	F1	Recall	Precision	F1	Recall	Precision	F1
FL	0.72	0.70	0.71	0.66	0.62	0.64	0.74	0.29	0.42
DPFL	0.69	0.68	0.68	0.26	0.22	0.24	0.20	0.24	0.22

Performance Comparison. We compare the performance of the S2P NILM model in federated and DP-federated learning setup. As stated, the federated approach provides an upper bound on the model's accuracy since, in DPFL, the addition of noise reduces performance. Table 8 shows the model predictions for on/off states for federated learning and DPFL. We observe that the model does well in disaggregating fridge energy signatures. Even within the privacy-preserving setup, the model still performs well with a marginal reduction in F1 score from 0.71 to 0.68. However, in comparison, the model accuracy drops significantly



(a) Fridge

(b) Microwave

(c) Kettle

Figure 15: Training loss of FL (non-DP) and DPFL models for different communication rounds.

for the microwave and kettle (66% and 47% respectively). This drop is presumably due to the distinctiveness of microwave and kettle power signatures when compared to the fridge. We hypothesize that microwaves and kettles reveal more about individual behavior as their use routines depend on user behavior. As DPFL tends to hide the influence of rarer information, it likely prevents the model from learning such distinct information.

Figure 15 depicts the training loss for different as the number of communication rounds increase. The figure shows that federated learning approaches converge much quicker than DPFL approaches. This is because no additional noise is added to the update parameters, resulting in faster convergence.

Impact of privacy parameter (ϵ). We evaluate the impact of ϵ parameter on disaggregating the appliance-level energy. Figure 16 shows the mean absolute error (MAE) between the appliance's mean power consumption and the models' prediction for varying *epsilon* values.

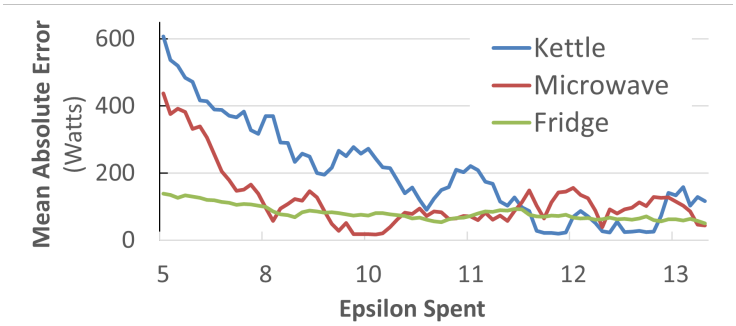


Figure 16: Mean Absolute Error from actual consumption for different privacy budget

Consider that the Kettle consumes a mean of 700W, the Fridge 200W, and the Microwave 500W. We note that a higher *epsilon* value denotes a relaxation of the privacy budget. In other words, the lower the epsilon, the stricter the privacy requirement, resulting in less information shared with the centralized server. As expected, the model's accuracy in predicting appliance energy usage improves as we relax the privacy constraints. In particular, we observe that the model tends to converge when $\epsilon > 12$.

Impact of privacy attacks. Next, we compare the performance of FL and DPFL in privacy attacks. As discussed, we use the Attacker Advantage (AA) metric to determine the attacker's success in discerning whether a client participated in the training. Note that AA relies on a binary classification (i.e., it was used to train the model or not), which is calculated as the difference between the True Positive Rate and the False Positive Rate; hence, we want the attacker to have a negative AA which means a higher False Positive Rate. Figure 17 depicts the AA for varying *epsilon* budgets. We observe that in FL, the attacker advantage is mostly positive, resulting in some success in discerning training participants (i.e., $TPR > FPR$). However, we note that in DPFL, $FPR > TPR$ for lower ϵ values, resulting in a negative attacker advantage. This shows a reduced success rate for an attacker while discerning the members in the dataset. However, when ϵ value increases, the TPR value becomes greater than FPR, indicating an improvement in the success rate in

identifying members in the dataset.

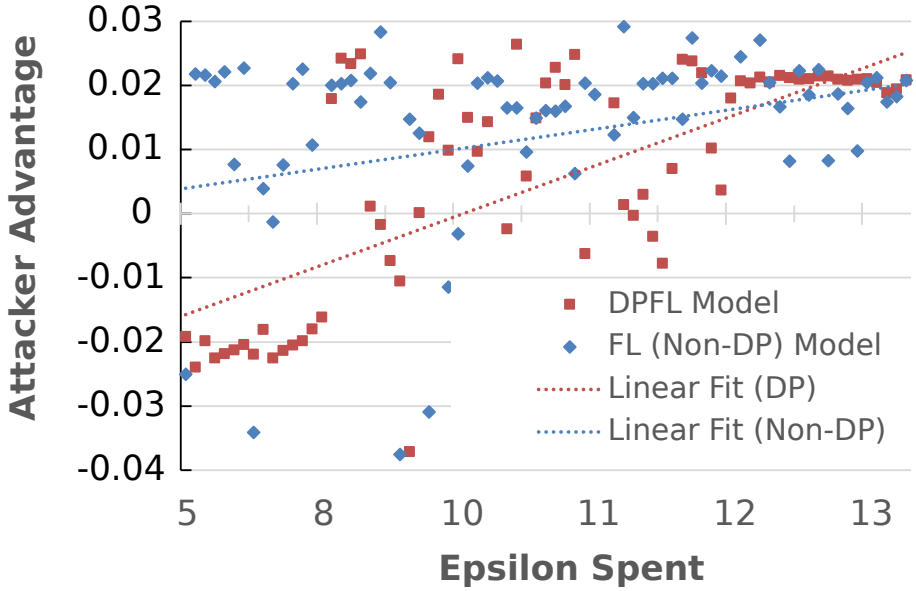


Figure 17: Attacker Advantage for the fridge trained model

As shown by Figures 16, 17, the Fridge can tolerate a lower Epsilon budget, which is within the range of highest protection against Attackers' Advantage (*epsilon* of 5-8), which indicates that the user can have better privacy if he is willing to only extract the power from select appliances.

In summary, DPFL can mitigate attacks for small ϵ values but provides similar privacy leakage compared to FL for higher values of ϵ. In particular, the attacker advantage in DPFL is similar to the FL when $\epsilon > 12$. This implies that, in practice, the ϵ value should ideally fall within the range of 5 to 8 to increase the attacker's false positive rate, where the attacker is more likely to misclassify data points that are part of the model's training set.

4.5 Conclusion

In this Chapter, we made strides towards answering $RQ2$, and we designed, implemented, and evaluated a framework to study the effectiveness of DPFL techniques on NILM models. This framework builds upon existing tools such as the NILMTK, Tensorflow Federated, NILM performance metrics, and known privacy attacks to provide a training framework for private NILM models. We evaluated our framework on two datasets and showed that the DPFL model is more robust at thwarting privacy attacks than non-private FL models. However, DPFL models have lower accuracy than FL, but we show that different appliance power consumption inferences can be more resilient to DP noise.

Finally, we also posit future research directions in developing models that are more robust to differentially private frameworks since we found that some models did not converge during DPFL training.

5.0 Feature-Driven Privacy-and Utility-Aware Obfuscation: Targeted Obfuscation of Human Voice

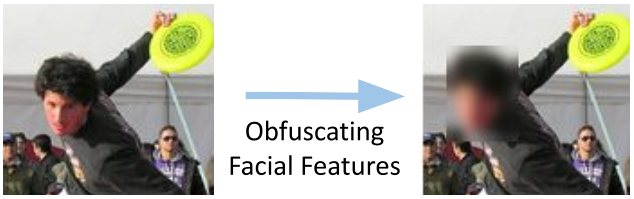
5.1 Introduction

Building on the foundations laid in Chapter 3, where we extended the traditional Utility-Privacy (UP) problem to incorporate Energy considerations, and in Chapter 4, where we explored the application of Federated Learning with Differential Privacy for Non-Intrusive Load Monitoring (NILM) settings, we refine a data obfuscation approach to answer $RQ3$ defined in Section 1.2. We apply privacy solutions to specific pieces of the data, rather than the complete data, with a goal to maintain high utility, control privacy loss, and make the models fit in resource-constrained IoT devices.

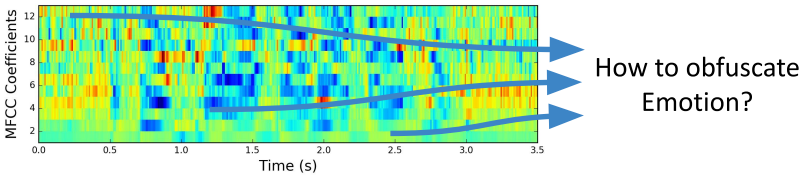
In Chapter 3, we evaluated different blurring effects as privatizers on images. Notably, we also evaluated Face Blur, where we specifically blurred only the facial region to minimize the blurring effect's impact on people counting task. We noted that it was better for utility and privacy, as well as energy consumption, allowing more computing-intensive effects since only a subset of the data had to be modified. Given the potential to improve privatizers' utility and privacy while consuming less energy, as intended by $RQ3$, we aimed to develop a more generic method to target the sensitive contents of data. In other words, by discerning which parts of the data have a higher bearing on Utility tasks and which are more associated with Privacy concerns, we can develop more effective privatizers that will consume less energy.

Our goal is to better obfuscate data, that is, distort or remove sensitive information from the data is to be shared with potentially malicious actors, without compromising its utility. These transformations must be designed carefully to ensure that they do not introduce prohibitively high energy demands to align with the principles of the Utility-Privacy-Energy (UPE) model introduced in Chapter 3 and respect the IoT-constrained environment discussed in Section 3.2.3.

In this chapter, we will highlight the development and implemention of neural network-based data transformations designed to obfuscate human voice data while remaining cog-



(a) Obfuscating faces in human photos



(b) What features are important for emotion in MFCCs

Figure 18: How to obfuscate specific data for arbitrary tasks when the important features are not evident?

nizant of the Utility, Privacy, and Energy trade-offs. By shedding light on how AI can help control these trade-offs more effectively, this chapter contributes to the broader discourse on privacy preservation in IoT settings. We focus on human voice data since it is rich with personal information but has a more manageable size when compared to images. However, our approach could be extended to other data types.

As discussed in Chapter 2, prior work has proposed data obfuscation as an effective measure to safeguard data in cloud-based environments [115, 116, 124, 94, 30]. This approach involves applying transformations, such as adding noise, to the data before transmitting it to cloud services. Such sensitive data obfuscation makes it more difficult for unauthorized entities to extract meaningful insights or identify confidential details. Data obfuscation techniques are prevalent in images, where *selective blurring* is applied to specific regions (e.g., blurring faces to protect identities), leaving others intact. In Figure 18) [116], we illustrate (at the top) how blurring/obfuscating faces allows for easy hiding of a persons identity. However, it is hard to figure out by looking at a visual representation of audio data (at the bottom of the figure), which features are important to obfuscate, and how to obfuscate them, for different privacy requirements.

We introduce *PrivSpeech*, a novel feature-driven privacy- and utility-aware obfuscation mechanism designed to selectively obfuscate specific (privacy) features while preserving the utility-serving features. Additionally, since the obfuscation process can potentially impact the utility of the data, as some privacy-specific features overlap with the utility-specific features, our obfuscation neural network (NN) is trained to restore data utility while effectively mitigating attacks.

PrivSpeech operates independently of any changes to the service provider because the model is locally applied before data is released from the IoT device. Other solutions that use NN masks to remove sensitive information change all input features, regardless of whether input features contain private information [88, 120, 16]. Also, by targeting only the sensitive subset of all the data, we can further minimize the PrivSpeech NN model size, which saves in computation (minimizing energy consumption) and allows it to fit devices with smaller memory capacity. Thus, current solutions cannot control whether utility data is also modified. Furthermore, we introduce an analysis of the various tradeoffs in selecting the top sensitive features and the implications for privacy and utility loss as additional features are obfuscated.

We test PrivSpeech on various combinations of voice utility and privacy tasks to show that it can preserve the signal utility (e.g., speaker identification) while removing sensitive information (e.g., emotion and gender); in other words, PrivSpeech **preserves the utility and protects privacy**. Since our privacy solution only obfuscates privacy-sensitive features, the utility signals remain unmodified, and we retain much of the original input. We demonstrate that by modifying only the privacy-sensitive features, we can control how much utility or private information is perturbed while maximizing the utility and privacy aspects of the data. Moreover, by targeting a portion of the critical features in our obfuscation strategy, we can train simpler and more compact NN models that can be better optimized to run on constrained IoT devices.

5.2 Background & Problem Statement

This section discusses our voice privacy risks and the threat model and specifies the problem statement.

5.2.1 Privacy Risks in Voice Data

Human speech involves various physiological processes in the production of voice. These processes encompass the vibrations of the vocal folds, which generate the fundamental unit of speech known as a phoneme, and the coordinated movements of multiple organs like the lips, tongue, and jaw that can produce distinct voice attributes [52, 130]. In the context of privacy, it is essential to note that there is a correlation between how voice is produced and several other physiological phenomena in the human body, such as emotion, body position, and diseases. Moreover, as the voice travels from the speaker to an audio sensor, information about the speakers' environment, such as the room size, can also leak [130].

The aforementioned voice-enabled services, when combined with other data, enable extraction of information such as emotions, behaviors, and even medical conditions, that can be used for personalized recommendations [130, 62] but also raise concerns regarding user privacy (e.g., targeted advertising and behavioral manipulation [16]). This kind of analysis amplifies the risks associated with voice data collection.

Data breaches and leaks of voice data from major platforms, such as Google and Alexa, also bring attention to the potential risks associated with collecting and managing voice data [60, 62, 132]. Unintentional disclosures or unauthorized access to data collected raise concerns regarding the potential misuse or abuse of personal information within the recordings. Voice recordings often include sensitive details, such as conversations, personal interactions, or private information shared during voice interactions with these platforms. Unauthorized access to such voice recordings can result in privacy breaches, identity theft, or other malicious activities.

5.2.2 Threat Model

We consider Voice User Interfaces (e.g., personal voice assistants) that transmit voice data to cloud-based services for storage and analysis; once users send their data to these services, they lose control over how the data is stored or processed. Our threat model assumes that the service provider has access to the user's voice data, and there is a possibility of unintentional misuse or disclosure, thereby compromising user privacy. An "honest-but-curious" service provider legitimately accesses the data but may engage in unauthorized activities or fail to adequately protect it, potentially resulting in privacy breaches. For instance, intended or authorized extraction of personal attributes (e.g., emotions or gender) for purposes like targeted advertising or behavioral manipulation [52, 113].

Our work aims to protect users from unauthorized inferences drawn from their voice data (protect privacy) while still enabling this data to access legitimate services (preserve utility). Note that we assume to know the attacker neural network model.

Although our technique addresses the privacy of voice data, other potential attacks and privacy risks are beyond the scope of this work, such as developing obfuscation models for other data modalities. Examples include attacks targeting the underlying infrastructure, such as malicious manipulation of voice recognition systems or spoofing attacks. While these concerns are significant, we focus on protecting sensitive attributes within the voice data and providing users with control over its usage and privacy implications.

5.2.3 Problem Statement

We consider multiple adversaries aiming to extract information from voice data while the user intends to distribute the voice data for a specific utility. Therefore, given a set of voice data as input, our objective is to identify the top-k sensitive features crucial for preserving privacy and minimizing the computing requirements by modifying less data. We then aim to develop an obfuscation mechanism that effectively protects user privacy by obfuscating these identified sensitive features while still retaining the utility of the voice data. It is important to note that the set of top-k sensitive features may vary depending on the specific privacy inference task. Thus, our challenge lies in determining an efficient method for accurately

identifying the top-k features to be obfuscated. Ultimately, we aim to find a comprehensive solution that effectively balances privacy with minimal computational requirements while retaining utility by appropriately obfuscating the top-k-sensitive features.

5.2.4 Privacy Feature Selection

Explainable AI was developed to address the lack of transparency in AI models to better understand errors, biases, and unfair outcomes, particularly for sensitive domains such as healthcare. For privacy, we can use explainable AI techniques to properly isolate the most important features for a sensitive inference that interests the user.

In the Explainable AI domain, Explainable Machine Learning (Explainable ML) algorithms have gained significant attention in recent years. These algorithms generate explanations to describe which dataset features are considered more relevant for predictions. They can be categorized as either *model-specific* or *model-agnostic*. *Model-specific* explanation algorithms only work on specific model architectures and use the internal parameters of the model architecture to provide explanations. In contrast, *model-agnostic* algorithms do not make any assumptions about the underlying model architecture and can be applied to different ML models to derive explanations. These algorithms treat the model as a black box and only require the input and the model outputs to provide explanations.

This work explores two model-agnostic approaches: SHapley Additive exPlanations (SHAP) and Principal Component Analysis (PCA). SHAP leverages Shapley Value from cooperative game theory to estimate the contribution of each feature in model prediction [93]. The algorithm creates different coalitions, representing a power set of all possible combinations of input features. Then, for all possible combinations, it computes the average marginal contribution of each feature by taking the difference between the prediction when the feature is present and absent in the coalition. This average marginal contribution is known as the Shapley Value, which measures the feature's importance in the model prediction.

Principal Component Analysis. PCA is often used to re-map the feature space by linearly transforming correlated variables into fewer uncorrelated variables. The linear transfor-

mation is done by projecting the original data into the reduced space using the eigenvectors of the principal components (i.e., covariance/correlation matrix). The resulting projected data is a linear combination of the original data that captures most of the variance in the data. Classifying each feature by the corresponding magnitude of the eigenvectors is possible, where a higher magnitude means higher importance.

SHapley Additive exPlanations. Formally, let $v : 2^n \mapsto \mathbb{R}$ be a coalition game and returns a value for each coalition $S \subset D$, where $D = \{1, \cdots, d\}$ represents a set of players. Then, the Shapley Value of player i for a coalition game v is given by:

$$\phi_i(v) = \frac{1}{d} \sum_{S \subset D \setminus i} \binom{d-1}{|S|}^{-1} (v(S \cup i) - v(S)) \tag{11}$$

To understand the importance of each feature in model prediction, the coalition game v can be cast as a feature attribution problem to compute model explanations by representing an individual prediction's dependence on different features. For a given model f, the value function for feature attribution on input x can be defined as

$$v_x(S) = \mathbb{E}[f(x_S, X_{D \setminus S})] \tag{12}$$

where $x_S \equiv \{x_i : i \in S\}$ represents a feature subset and X_S is the corresponding random variable.

Shapley value is difficult to calculate because they require computing a feature's contribution to all possible combinations, with an exponential run time in the number of features. Since the exact computation of Shapley values is computationally infeasible, prior work has used approximations to help summarize each feature's contribution to model prediction. In particular, we use SHAP proposed in [93] that estimates the Shapley Value by viewing it as a weighted least square problem.

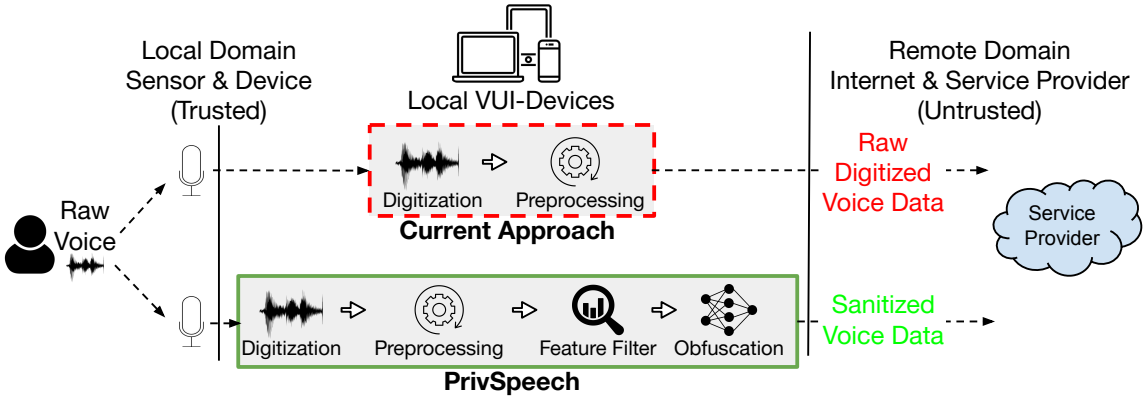


Figure 19: PrivSpeech System Overview.

5.3 PrivSpeech Design

We discuss our approach's overall design and then describe the details of our utility-preserving voice data obfuscation framework.

5.3.1 System Overview

Figure 19 depicts the overall approach of PrivSpeech in comparison to existing techniques; both have three entities: the user, a voice-enabled device equipped with a microphone, and the service provider. In existing approaches (see middle top part of the figure), the voice input captured from a local microphone is digitized and processed. The resulting raw or processed data is subsequently transmitted to an external service provider for further analysis and processing to deliver desired services, such as speaker verification. Within this deployment model, the voice sensor and processing devices are considered trusted and local to the user. However, as the raw digitized voice data is transmitted to the service provider, potential risks and vulnerabilities arise, posing threats to the exposure of sensitive content within the voice data.

In contrast to current techniques, PrivSpeech preserves the privacy of sensitive attributes by selectively obfuscating, on the local device, only the features related to the sensitive task.

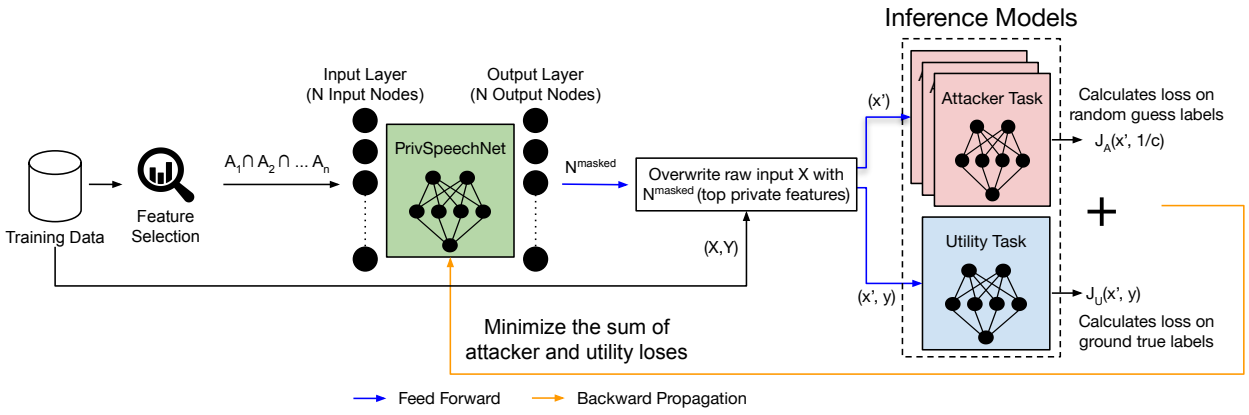


Figure 20: PrivSpeechNet Obfuscation Model Training Framework

The primary objective is to ensure that only essential information is shared with the service provider, empowering users to maintain control over their data and strike the desired balance between utility and privacy. A key design goal of PrivSpeech is to achieve efficient obfuscation by selectively perturbing minimal data, enabling its execution even in resource-constrained environments. This approach empowers users to control the sensitive features (identified by PrivSpeech) instead of dealing with all features indiscriminately (as done in previous works). We will show that selective obfuscation minimizes the computational requirements for data obfuscation, making it feasible for implementation when resources are limited, as in IoT devices.

PrivSpeech comprises two key components: (i) *Sensitive Feature Selection* and (ii) *PrivSpeechNet Obfuscation Model*. The Feature Selection component plays a crucial role in identifying the important features of voice data to maintain privacy and utility. By accurately identifying these features, the subsequent obfuscation model will only modify these unique features to balance privacy preservation and utility retention. The *PrivSpeechNet Obfuscation Model* is designed to address the challenge of perturbing selected sensitive privacy features, while leaving the remaining ones unaltered, minimizing the impact on utility. This approach allows for a fine-grained data adjustment, maintaining the delicate tradeoff between privacy and utility in a computationally-efficient manner.

5.3.2 Sensitive Feature Selection

The Sensitive Feature Selection component in PrivSpeech aims to determine which features mostly contribute to sensitive inferences and utility tasks. PrivSpeech employs two model-agnostic approaches, namely Principal Component Analysis (PCA) and SHapley Additive exPlanations (SHAP), to achieve this goal [150].

PCA is used to identify the data's most important information/components/features according to how each feature contributes to the variance explained by each principal component. By analyzing the contributions of these principal components, PrivSpeech can sort the top-k most important features based on the weight of the principal component contributions and target only those with the highest contributions.

Similarly, we use SHAP feature contribution estimations for a model's prediction to sort the importance of features. However, while PCA is only based on the data, SHAP is model dependent, which means that SHAP captures the feature importance related to particular inferences and can be used to sort features related to specific sensitive inferences of the attacker model.

Overlapping top-k features. SHAP provides model dependent top-k features; in other words, for each inference task, SHAP will output the top-k most important features. Since there are multiple tasks (the one the user is requesting and the various attacks), we may have multiple attacker models intersecting each other's top-k set. In such cases, we must determine which top-k features to use to maximize privacy. This does not happen for PCA, which is only a remap of the data and does not change according to any model.

We use SHAP to illustrate how feature selection is performed in PrivSpeech. Consider a training dataset, denoted as $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, which contains sensitive information. Here, x and y represent vectors in $\mathbb{R}^c$, and $(v_1, v_2, \ldots, v_c)$ are the elements of vector x. The value of c corresponds to the dimension of the input data, representing the number of features. Let $S = \{z_1, z_2, \ldots, z_n\}$ represent the set of Shapley values obtained using a function $g(M, x_i)$ that calculates the Shapley values z_i for each $x_i \in D$ of a model M that represents an attacker model. Note that PrivSpeech utilizes Shapley values to isolate the sensitive portions of data. By analyzing the Shapley values z_i, one can identify

which features significantly impact the model's predictions for a given input x_i and are thus considered sensitive. To select the top-k features for a given model, we summarize across all inputs as follows.

$$z' = \frac{\sum_{i=1}^{n} |g(M, x_i)|}{n} \tag{13}$$

where z' represents the average Shapley value of feature i over all n inputs. We then use the z' values to sort the features for each task (i.e., extract the most important features for utility and for all adversary models M).

We use the following process to merge the top-k important features from each model. Let z'_1 and z'_2 represent two lists of important features sorted in descending order. To combine these lists while preserving their order, we take their union and truncate the resulting list to the top-k elements. Mathematically, this operation is denoted as $z'_3 = z'_1 \cup z'_2[:k]$. Using our obfuscation model, these merged features are then utilized to identify the most important attributes in each data instance, which are filtered and obfuscated.

5.3.3 PrivSpeechNet Obfuscation Model

In this step, the sensitive features obtained in the previous step are obfuscated so that the adversary model cannot infer the specific private property; in other words, our goal is to ensure that the adversary model performs no better than a random guess (which is similar to having a really inaccurate model), while maintaining the high performance of the utility tasks. To achieve this, PrivSpeechNet employs a NN-based approach, represented by $H(D;\theta)$, where θ represents the weights.

In PrivSpeech, the utility to the user is represented by the set of utility models M_u, which may include tasks like speaker verification. On the other hand, the set of adversary models M_a aims to extract sensitive information from the data D. PrivSpeechNet aims to minimize the loss function $\mathcal{L}$, a combination of the losses on M_u and M_a.

$$\mathcal{L} = \alpha \mathcal{L}_u + \sum_a \omega_a \mathcal{L}_a \tag{14}$$

where $a \in A$ denotes the adversary models, α and ω_a are weight parameters that control the importance of each component. Further, $\mathcal{L}_u$ and $\mathcal{L}_a$ denote the utility and adversary loss, respectively. To compute the adversary loss $\mathcal{L}_a$, for each pair (x, y), the true label y is substituted by $1/c_a$, where c_a is the number of classes[1] for the respective adversary model M_a. During backpropagation, the obfuscation model H will learn to transform the data to force adversary model M_a to predict a uniform random distribution based on the number of classes c_A. This substitution ensures we can simply minimize the loss while training the model H. Thus, the loss function $\mathcal{L}$ encourages the model to balance between preserving utility $\mathcal{L}_u$ and limiting the effectiveness of adversary models $\mathcal{L}_a$ by assigning appropriate weights α and ω_a to each component.

During the training process, PrivSpeechNet learns to obfuscate the sensitive attributes present in the input data x by modifying it through the neural network H. As shown in Figure 20 The network's weights θ are adjusted iteratively to minimize the loss simultaneously on both the utility models M_u and the adversary models M_a. This training objective ensures that the obfuscated data maintains its utility for the user while making it difficult for the adversary to extract sensitive information.

Algorithm 1 outlines the pseudocode for training the model. The algorithm begins by calculating the utility and adversary feature importances from the dataset D using the utility model M_u and the adversary model M_a, respectively. These importances are combined, and the top-k features, representing the most relevant attributes, are selected. Subsequently, the original dataset D is filtered based on these top-k features, resulting in a modified dataset D' where sensitive data has been removed.

The model is then trained using the D' dataset. For each instance in the dataset, the model applies an obfuscation model H to generate an obfuscated mask δ, which is then applied to the top-k features of the input. It then computes both utility and adversary losses. The utility loss is determined by comparing the model's output on the obfuscated instance with the utility ground truth y, using the utility model M_u. Similarly, the adversary loss is calculated by iterating over each adversary model in the set A, determining the number

[1]As mentioned above, we want the adversary inference to be as good as guessing randomly, which has a success rate of $1/c_a$.

of classes in the adversary model, and utilizing this information along with the obfuscated instance and the respective adversary model. The adversary losses across all models in A are summed to obtain the total adversary loss.

Finally, a joint loss $\mathcal{L}$ is computed as a weighted sum of the utility loss and the total adversary loss. By applying backpropagation, the obfuscation model H parameters are updated based on this joint loss. This iterative process is repeated for the specified number of epochs, training the obfuscation model to simultaneously maximize utility preservation and minimize the risk of information leakage that adversaries could exploit.

Algorithm 1 PrivSpeechNet Model Training Pseudocode

procedure TRAIN_PRIVSPEECH($H, D(x, y), M_u, M_a, k$)
  $Z' \leftarrow$ feature_importance(M, D) $\forall M$ ▷ compute feature importance for all models
  $z' \leftarrow$ merge_select_top-k(Z', k)
  $D' \leftarrow$ remove_sensitive_data(D, z') ▷ based on top-k
  for $e \leftarrow$ *epochs* **do**
    for $x, x', y \leftarrow D, D'$ **do**
      $\delta \leftarrow H(x')$ ▷ Generate obfuscation mask
      $x_{obf} \leftarrow$ obfuscation(x, δ, z') ▷ Obfuscate top-k
      $\mathcal{L}_u \leftarrow$ calc_loss($M_u(x_{obf}), y$)
      for $a \leftarrow A$ **do** ▷ For each adversary model
        $c_a \leftarrow$ number_of_classes(a)
        $\mathcal{L}_a \leftarrow$ calc_loss($M_a(x_{obf}), 1/c_a$)
        $\mathcal{L}_a^{total} += \omega_a \mathcal{L}_a$
      end for
      $\mathcal{L} \leftarrow \alpha \mathcal{L}_u + \mathcal{L}_a^{total}$
      backpropagation($H, \mathcal{L}$)
    end for
  end for
end procedure

Energy Consumption Perspective Note that while PrivSpeechNet is trained to optimize Utility and Privacy according to the select top-k set of sensitive features, the size of k will dictate how much of the data has to be obfuscated which can increase the computing requirements of PrivSpeechNet when deployed. The tradeoffs UPE tradeoffs for different sizes of k are discussed in the evaluation Section 5.5.

Adversarial retraining A key strength of our approach is controlling how sensitive features are perturbed, allowing users to have a say in the obfuscation process. However, it is

important to note that adversaries can still attempt to retrain models using the obfuscated data. Prior work has shown that in such cases, privacy inference attacks can still be carried out [131]. PrivSpeech incorporates a two-step obfuscation strategy to mitigate against retraining-based attacks.

Primarily, we remove the top-k sensitive features by setting their values to 0, effectively removing them from the data. Next, we train PrivSpeechNet to modify the remaining features, which will, in effect, attempt to re-encode utility information that might have been accidentally removed when we set the top-k features to 0. In this scenario, the value of k determines the number of sensitive features removed, which is effectively controlling the amount of information transmitted to the service provider. A higher value of k enhances privacy by preventing adversaries from performing well even after training on obfuscated data. However, removing too many features may negatively impact utility. We extensively analyze this tradeoff in our evaluation presented in Section 5.4, demonstrating the effectiveness of our approach against adversaries with retraining capabilities.

5.4 Experimental Setup

This section provides an overview of the three datasets used in our experiments and describes the experimental setup, including the inference models, baseline algorithms, and evaluation metrics.

5.4.1 Datasets

In our evaluation, we utilized three datasets: RAVDESS, EmoDB, and EMOVO, widely recognized and utilized in human voice emotion-related research [90, 37, 1]. While these datasets were originally designed for emotion-related tasks, they also provide sufficient information for gender identification and speaker verification tasks, as the actors' identities and genders are specified. Moreover, these datasets have recordings in multiple languages, namely German, English, and Italian, offering a diverse linguistic context.

RAVDESS is the Ryerson Audio-Visual Database of Emotional Speech and Song; our largest dataset has 24 North-American actors, 12 males, and 12 females. The actors recite various lines expressing eight different emotions. The dataset consists of 7,356 labeled instances, having both speech and song recordings in audio and video formats. RAVDESS stands out for incorporating auditory and visual modalities, providing a unique resource for studying emotional expression.

EmoDB is the Berlin Emotional Speech Database, a smaller yet highly focused dataset comprising 800 sentences spoken by ten German-speaking actors, each delivering the sentences in seven distinct emotional states. EmoDB emphasizes emotional variance and consistency of delivery, offering a valuable resource for investigating emotion-related tasks.

EMOVO is primarily used for Italian speech emotion recognition. It provides a multilingual perspective by including recordings in multiple languages. EMOVO consists of 588 utterances performed by 14 actors, each portraying seven emotions. EMOVO highlights the influence of cultural nuances and native language on emotional expression.

5.4.2 Experimental Setting

For evaluating PrivSpeech, we deployed our system on a Raspberry Pi 4 (RPi4) as our IoT device. The RPi4 has a Cortex-A72 (ARM v8) CPU with four cores and 4GB of RAM, suitable for implementing NN models using TensorFlow Lite. The RPi4 represents a typical home voice assistant IoT device with limited computational capabilities, making it a relevant platform for evaluating the efficiency and practicality of PrivSpeech in real-world scenarios.

5.4.2.1 Models

Using the Ravdess, EmoDB, and EMOVO datasets, we trained three fully connected NN models for gender classification, emotion classification, and speaker verification (see details of each of these tasks below). Our evaluation used these trained models as utility models or attacker models.

To train the models, we performed pre-processing on the voice data. We extracted 120

Mel Frequency Cepstrum Coefficient (MFCC) features from each voice recording. These coefficients capture essential characteristics of the voice signal, such as timbre, pitch, phonemes, and spectral shape, which are linked to the articulatory configuration of the vocal tract [130]. Additionally, we augmented the data by calculating the delta of each MFCC (i.e., the difference between each neighboring MFCC component), resulting in a total of 240 features per recording. In all models, (a) extensive manual search and experimentation determined the NN hyperparameters for promising results; (b) to prevent overfitting, a dropout rate of 20% was applied after each layer, and (c) 80% of each dataset was used for training and 20% for testing.

Gender classification model: The goal of this model was to classify the gender of the speaker as male or female. After extensive optimization, we settled on a fully connected NN with six hidden layers containing 192, 160, 128, 96, 64, and 32 neurons, respectively, with a swish function as activation [119]. Also, we used the Adam optimizer with a learning rate of 1e-4. We trained the model for 5,000 epochs with a batch size of 128. This model achieved an F1 score of 1 on the test dataset for each of the three datasets.

Emotion classification model: The emotion classification model addressed the more complex task of inferring emotions from voice, considering the nuanced nature of emotional information that can impact the duration and intonation of speech. We developed a NN specifically for emotion classification and trained it on the respective datasets, which varied in the number of emotion classes (ranging from 7 to 8 depending on the Dataset).

Our chosen architecture consisted of a fully-connected NN with seven layers, each comprising 512 neurons. We employed the *selu* activation function in the first layer, *gelu* in the intermediary layers, and *tanh* in the final layer. The optimization function utilized a varying learning rate, initially set to 1e-4, based on Cosine Decay with restarts. The model was trained for 10,000 epochs with a batch size of 180. Our simple, fully connected neural network achieved an F1 score of 0.87 on the Emodb dataset, 0.80 on the EMOVO dataset, and 0.83 on the Ravdess dataset. These scores are not far from state-of-the-art models in emotion detection, despite our use of a simpler neural network model [143, 140, 137]. One advantage of our model is its efficient training and evaluation process.

Speaker Verification model: We developed a deep NN model to classify and verify the identity of users. The model was trained to recognize the unique voice signatures of each actor in the dataset.

For this task, our NN had six layers, with sizes of 192, 160, 128, 96, 64, and 32 neurons, respectively. We employed the *selu* activation function for the first layer, *swish* for the intermediate layers, and *tanh* for the final layer. We used the Adam optimizer with a learning rate of 1e-4 to optimize the model. The model was trained for 1,000 epochs with a batch size of 180. During training, the model achieved an F1 score of 1 on the test set for each dataset, indicating perfect performance in verifying the identity of users.

5.4.2.2 Feature Selection Strategy

Our evaluation of PrivSpeech compared several feature selection strategies for privacy preservation. These strategies are:

- **ShapRev (Reverse Shap)**: This strategy selects the top-k features based on the reverse order of their Shapley values. It investigates whether selecting non-important features contributes to privacy preservation while retaining utility.
- **RND (Random)**: This strategy randomly selects k features without any specific criteria. It serves as a baseline to compare against other selection methods.
- **PCA (Principal Component Analysis)**: This strategy selects the top-k features based on PCA. By selecting the top-k principal components, we aim to capture the most important information while reducing dimensionality for privacy preservation.
- **NonUtil (Shap)**: This strategy selects the top-k features based on their Shapley values but focuses on preserving utility by only obfuscating the non-utility-related features. This can minimize the performance loss on the utility model since it guarantees that the top-k utility features remain untouched after obfuscation.
- **TopK (Shap)**: This strategy selects the top-k features based on their Shapley values. Shapley values quantify the contribution of each feature to the model's output.
- **Removal (Shap)**: The top-K features selected by Shap are set to 0. We use this approach as a baseline for comparison since removing features will often degrade the

utility model performance.

5.4.2.3 Adversaries

In our evaluation, we assess the performance of PrivSpeech against the following adversaries:

- **Static Adversary**: The adversary does not retrain the model but attempts to infer private information from the obfuscated data (i.e., with obfuscated top-k features). In this setup, the top-k features are obfuscated with PrivSpeechNet, and the remaining features are not modified.
- **Dynamic Adversary (DA)**: We consider a dynamic adversary capable of retraining their model using obfuscated data. To address this scenario, we modify PrivSpeech's obfuscation strategy as discussed in Section 5.3.3 and explore three different approaches:
  - **DA-TopK**: we perform the same obfuscation strategy against a static adversary, but we show the adversary's performance after retraining the model on the obfuscated data.
  - **DA-Rest**: change the top-k features to zero, effectively removing all sensitive information. The remaining features are used as input to train PrivSpeechNet as well as modified during obfuscation. This variation examines the effectiveness of PrivSpeech when only a subset of features are used for training the model, with the top-k sensitive features removed. This limits PrivSpeech to only learn from the top-k most important sensitive features, which reduces the risk of adding other information during obfuscation.
  - **DA-All**: similar to DA-Rest; however, all the features are used to train PrivSpeechNet instead of only the remaining ones after setting the top-k features to zero. This analysis explores the performance of PrivSpeech when all features are employed to generate the mask while explicitly targeting the top-k features for obfuscation. This approach allows PrivSpeechNet to learn from all the data, which can increase the risk of adding sensitive information. However, it will often be able to restore better

the utility model that might have been too much affected by the removal of the top-k features.

5.4.2.4 Performance Metrics

Since we evaluate neural network models performing classification tasks, we used the F1 score as the performance metric to evaluate each model M, defined as

$$F_1(M) = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

This metric is often used in machine learning investigations, because it provides a balanced assessment of precision (the ratio of correctly predicted positive observations in relation to the total predicted positives) and recall (the ratio of correctly predicted positive observations in relation to the ground truth). This provided a holistic view of model performance across these tasks, explicitly offering insights into how effectively each model identified true positives while minimizing false negatives and false positives. Moreover, the F1 score is resilient to imbalanced datasets since it is calculated based on the harmonic mean of both precision and recall.

Utility Score. We utilize the F1 score as a measure of utility, which ranges from 0 to 1. Higher F1 scores indicate better performance in terms of precision and recall.

Privacy Score. We evaluate the performance of our adversarial tasks using a modified F1 score metric given by the equation:

$$P_a = \left(1 - \frac{1}{c_a}\right) - \left|F_1(M_a) - \frac{1}{c_a}\right| \tag{16}$$

where P_a represents the success of the adversary task. This metric measures how well the adversary performs compared to a random guess ($1/c_a$). If the adversary's performance is close to a random guess, the value of $\left|F_1(M_a) - \frac{1}{c_a}\right|$ will be close to zero, and the privacy metric P_a will be close to the maximum value defined by $\left(1 - \frac{1}{c_a}\right)$. On the other hand, if $F_1(M_a)$ is 1, then P_a is 0, which means that the attacker is performing well and the privacy is 0 (i.e., no privacy). We then apply min-max normalization between $1/c_a$ and $1 - (1/c_a)$ to ensure the metric falls within a specific scale. This normalization procedure re-scales the

values of P_a to a range of [0,1], allowing a more straightforward interpretation.

Note on the use of entropy as a Privacy Score. While we could have used entropy as a measure of randomness for the adversary model performance, it would ignore how exactly the attacker is miss-predicting in relation to the input space. For example, consider a scenario where an adversary tries to classify images of different Persons correctly. If the attack results in a skewed distribution favoring a single person (low entropy), but that person is not correct, the attack fails despite the low entropy. Hence, entropy values not always relate to how well the model is predicting the negative or positive classes.

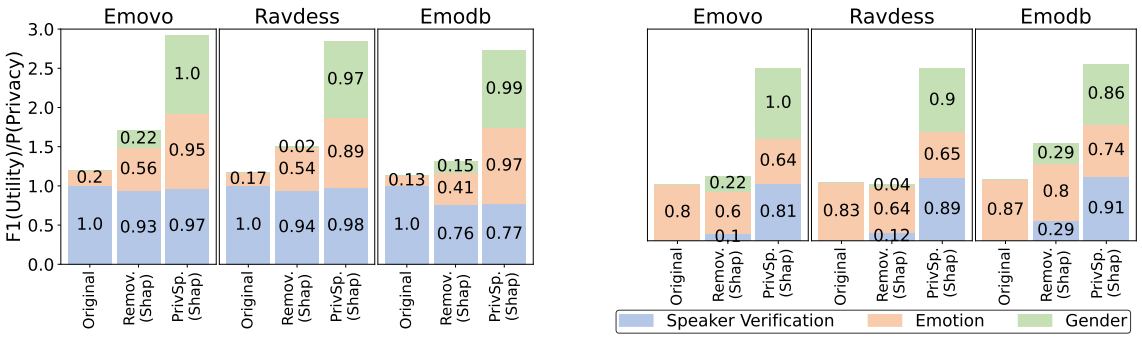
5.5 Results

In this section, we present the empirical evaluation results of PrivSpeech.

5.5.1 Baseline Performance

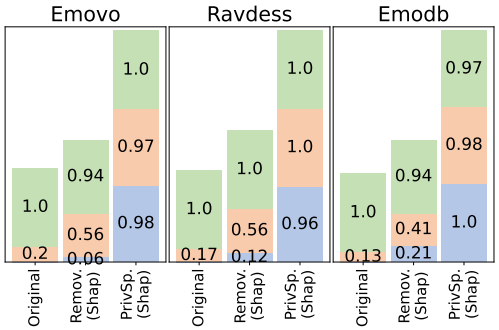
To evaluate the performance of PrivSpeech, we compare it to the original unmodified data and a simple approach of removing the top-30 (Shap) features from the data without employing PrivSpeech. Figure 21 illustrates the performance of PrivSpeech compared to these scenarios. The y-axis denotes the utility or privacy scores depending on which task is selected as a utility. The x-axis discerns the privacy solution method, further divided by each Dataset Evaluated. Note that all models' scores are stacked; no bar will show if the value is 0.

The privacy scores on the original unprotected data are significantly low (around 1.2, left bar for each dataset) for all datasets, indicating that the attacker can extract information. Additionally, simply removing the top-30 features had minimal impact on privacy. For example, as shown in Figure 21-(a), middle bar, orange portion, shows that the privacy scores for emotion privacy improved by 0.36, 0.37, and 0.28 for EMOVO, Ravdess, and Emodb, respectively, when the top 30 features were removed (note also the increase in



(a) Speaker Verification as Utility

(b) Emotion as Utility



(c) Gender as Utility

Figure 21: PrivSpeech performance with Top-30 features obfuscation.

gender privacy). We show that PrivSpeech (right bar) is able to preserve privacy and retain utility, achieving very high privacy and utility scores.

We further investigated the performance of PrivSpeech by switching the utility task with other tasks. Specifically, we used Emotion (or Gender) as the utility task and Gender (or Emotion) and Speaker Verification as the privacy task. As shown in Figures 21-(b) and 21-(c), we observed similar performance trends in both scenarios (i.e., an increase in privacy and utility), indicating that PrivSpeech is effective in removing sensitive features for different models while preserving high utility scores.

Key Observations: PrivSpeech demonstrates its ability to enhance privacy while preserving data utility through selective obfuscation of features. It can handle various tasks as privacy and/or utility, regardless of the selection or combination. By adapting to different

scenarios, PrivSpeech balances privacy protection and utility retention.

5.5.2 Feature Selection Strategy

We analyzed PrivSpeech's performance using different feature selection strategies, as shown in Figure 22. PrivSpeech utilized the top-120 features for the various privacy and utility setups across all datasets in this evaluation. The performance of each strategy was evaluated using an average score, which represents the average of the utility and privacy scores.

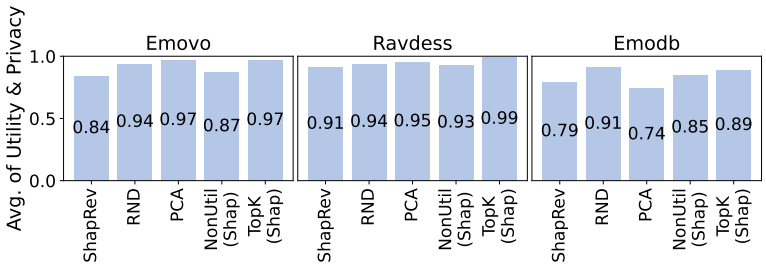
In Figure 22, for the EMOVO dataset, the PCA method only won when Emotion was Utility Figure 22-(b), having the score of 0.87, while TopK (Shap) had the second-best score, indicating that using important features is crucial for preserving privacy and retaining utility. ShapRev had the lowest score of 0.7. For the RAVDESS dataset, the TopK (Shap) method outperformed the others, scoring 0.88. The ShapRev and PCA methods had the lowest scores, each obtaining a score of 0.82. In the EMODB dataset, the TopK (Shap) method again achieved the highest score of 0.9.

Overall, the TopK (Shap) method consistently achieved high scores across all datasets, even when we switched tasks. This indicates that effectively targeting the top-k features for each model enables PrivSpeech to transform the data in a manner that minimizes both the impact of the adversary and the loss of utility. Conversely, the ShapRev method consistently yielded the lowest scores, as expected, since selecting the least important features reduces the model's information and control over the sensitive tasks. These results highlight the advantages of the TopK (Shap) strategy in achieving a balance between privacy and utility.

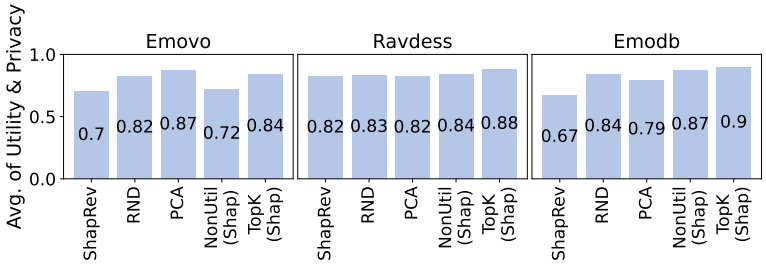
Key Observations: The SHAP approach, such as the TopK (Shap) method, which selects important features and performs targeted obfuscation on sensitive attributes, consistently achieved higher privacy and utility scores.

5.5.3 Preserving Utility Features

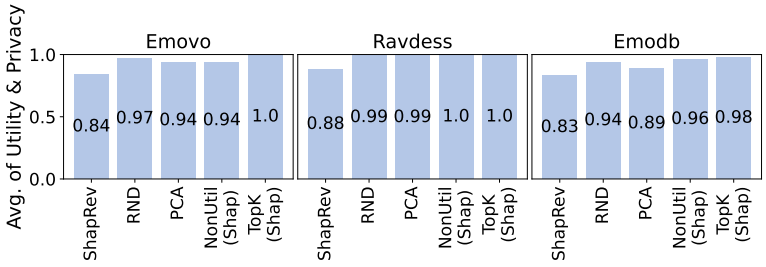
So far, we have analyzed the performance of obfuscating the top-k-sensitive features. We now analyze the performance for selecting the *non-util* features (i.e., obfuscate non-utility



(a) Speaker Verification as Utility



(b) Emotion as Utility

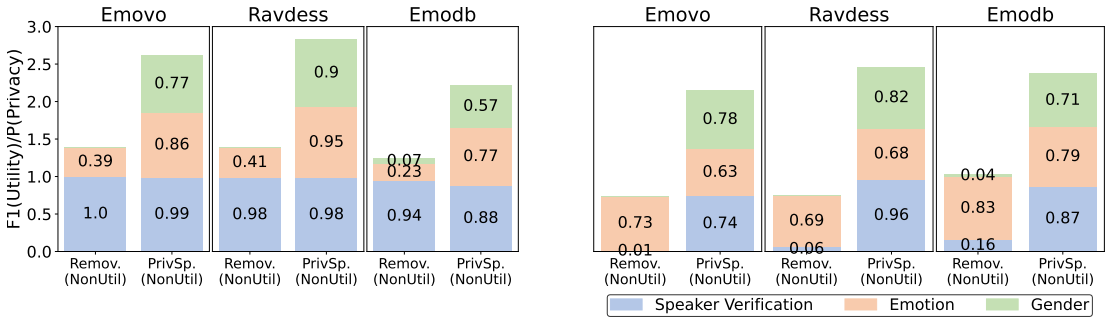


(c) Gender as Utility

Figure 22: PrivSpeech performance for different feature selection strategies when Emotion is Utility.

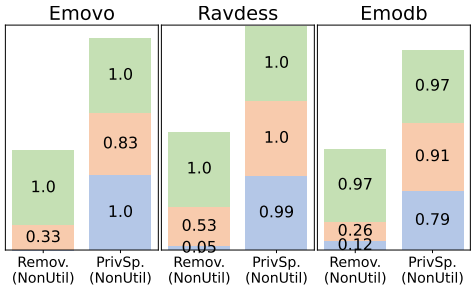
features, thus protecting utility-specific features). Figure 23 showcases the performance of PrivSpeech when protecting the top 120 most important features for utility while obfuscating the remaining features. We also evaluate a naive approach that preserves the top-k utility features but removes (rather than just obfuscates) the rest.

The results demonstrate that the naive approach of just removing the top-k fails to preserve privacy while achieving good utility scores. This can be attributed to the fact that although important utility features are retained, sensitive features are not targeted for



(a) Speaker Verification as Utility

(b) Emotion as Utility

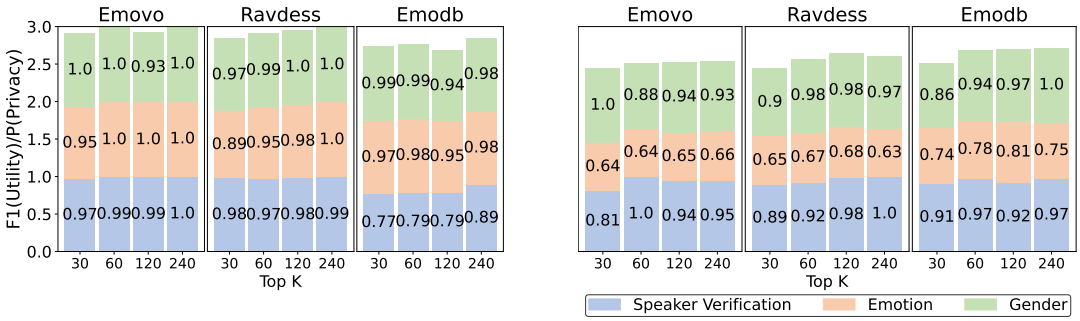


(c) Gender as Utility

Figure 23: Protecting the Top 120 Features

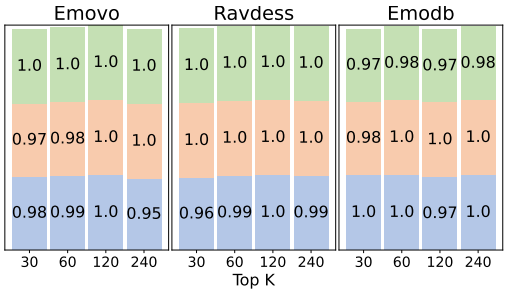
obfuscation, leading to low privacy scores. In contrast, PrivSpeech achieves high utility scores while simultaneously achieving high privacy scores across all datasets. By considering the importance of both utility and privacy, PrivSpeech effectively protects sensitive features while preserving the utility of the remaining features. These results highlight the advantages of PrivSpeech in achieving a balance between privacy preservation and utility retention.

Key Observations: Preserving the top-k utility features yields higher utility scores while sacrificing privacy. This highlights the trade-off between utility and privacy in the context of selective obfuscation. Additionally, our technique allows for control over the obfuscation process, enabling the preservation of utility if it is the primary objective.



(a) Speaker Verification as Utility

(b) Emotion as Utility



(c) Gender as Utility

Figure 24: Effect on performance as we obfuscate more top-k features using PrivSpeech.

5.5.4 Choosing top-k: sensitivity analysis on k

In Figure 24, we analyze the impact of varying the number of features considered for obfuscation in PrivSpeech. We observe diminishing returns as the number of features increases for the Ravdess (Speaker Verification as Utility) and Emovo and Emodb (Emotion as Utility). However, Emovo/Emodb (Speaker Verification as Utility) and Ravdess (Emotion as Utility) show minor tradeoffs. This indicates that PrivSpeech can efficiently force adversary models to no better than random while restoring the utility model by just targeting top-30 features when facing a static adversary. Notably, obfuscating the top 30 features allows for a smaller PrivSpeechNet while maintaining effectiveness, as discussed in Section 5.5.6.

Similar trends are observed when switching the utility and privacy tasks, shown in Figure 24(b) and 24(c). When obfuscating all features, the performance is similar to targeting

the top 60 features, suggesting that only a few (about 25%) features are necessary to preserve privacy and utility. Additionally, high privacy scores are often achieved, indicating that the privacy tasks' performance is no better than random guessing.

Key Observations: Our analysis shows diminishing returns as the value of k increases, indicating that selecting sensitive features to obfuscate is crucial for achieving a balance between privacy preservation and utility retention.

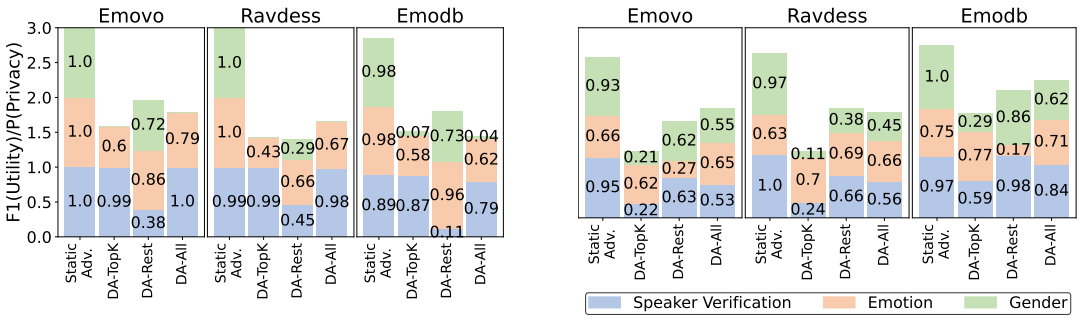
5.5.5 Dynamic Adversary

In this section we present the results for PrivSpeech if the adversary can retrain his model. Since PrivSpeech learns to "fool" the adversary models, it may still re-encode sensitive information which can be relearned if the adversary re-trains his model. In this scenario, PrivSpeech will remove the top-k private features and will target all the other features for obfuscation while preserving utility.

We now explore the scenario where the adversary has the ability to retrain their model using the obfuscated data generated by PrivSpeech. To address this, we modify the PrivSpeech obfuscation strategy to ensure no sensitive features are transmitted to the cloud setting the top-k features to zero, effectively removing them from the data. In these results, we show only results for $k = 220$, which we found empirically to have a good utility-privacy balance after extensive state space exploration.

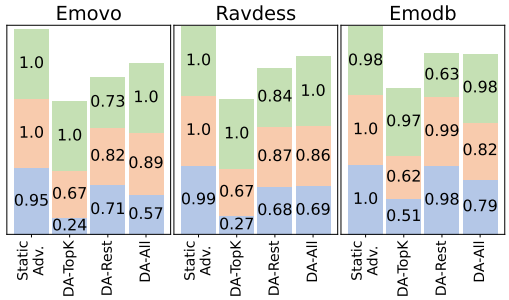
Figure 25 presents the adversary's performance with retraining capability. Each figure displays the results for the baseline approach without retraining (Static Adv.) and the dynamic adversary's performance using different obfuscation strategies: *DA-TopK*, *DA-Rest*, and *DA-All*. It is expected that privacy is compromised when the adversary can retrain their model. For instance, in Figure 25(a), when employing the *DA-TopK*, where PrivSpeech obfuscates with the default approach the Gender Privacy score becomes 0, indicating a complete loss of privacy when the adversary can retrain the model.

However, modifying the obfuscation strategy can achieve higher privacy or utility scores when adversaries can retrain their models, as demonstrated by the results of the *DA-Rest* and *DA-All* approaches. With *DA-Rest*, the top-k sensitive features are set to zero; the adversary



(a) Speaker Verification as Utility

(b) Emotion as Utility



(c) Gender as Utility

Figure 25: PrivSpeech performance when adversary does not (Static Adversary) and does retrains on obfuscated data (DA-*).

has little to no information to carry out privacy attacks, resulting in improved privacy scores. However, with *DA-Rest*, the model can only use the top 220 features, limiting the information necessary to restore the utility, as some utility features may have been removed. With *DA-ALL*, PrivSpeech will still remove the top-k; however, it will use all the features to train and modify the remaining 20 features, enabling a better utility recovery at the cost of losing privacy since private information may be re-added. This emphasizes the trade-off between privacy and utility, where users can prioritize one at the expense of the other.

Figure 25(b) presents the performance of the Emotion task as a utility with adversary model retraining. Similar trade-offs are observed, with the *DA-Rest* approach prioritizing privacy at the expense of utility. However, the *DA-All* approach achieves a more balanced protection strategy, although privacy scores are still lower than those without adversary

Table 9: PrivSpeechNet model performance for top-240 and model performance for different top-k values.

Model	**Instructions***	**Memory Reads***	**Memory Writes***	**Cache Misses***	**Runtime (ms)**
tflite	0.5627	0.1752	0.1113	0.0175	1.285
Normal	85.946	24.319	16.393	1.674	130
	top-k	**fp32 size (KB)**	**int8 size (KB)**	**Runtime (ms)**	
tflite	240	115.6	28.9	1.285	
	120	70.1	17.5	1.210	
	60	47.4	11.8	1.200	
	30	36.0	9.0	1.179	

* measured in Millions

retraining. In Figure 25(c), a similar trend is observed for the Gender task as a utility. However, we note higher privacy scores in this case.

Key Observations: *PrivSpeech can achieve privacy even when adversaries can (re)train their models on obfuscated data. However, this also affects the overall utility, highlights the challenges of protecting privacy against adversaries with retraining capabilities, and emphasizes the importance of carefully considering the trade-offs between privacy and utility when selecting the appropriate obfuscation approach.*

5.5.6 Model Resource Analysis

We assess how much it takes, in terms of CPU and memory, for PrivSpeech using the top-240 features and optimize our model using TensorFlow Lite (*tflite* requires an RPi4 device) compared with a regular TensorFlow implementation. Utilizing all 240 features as input to the PrivSpeech obfuscation model provides an upper bound of computational resources required for running the model. Table 9 presents the results of employing TensorFlow Lite model optimization. The tflite model shows fewer instructions and cache misses, resulting in a performance improvement of 100x compared to the regular TensorFlow model. This is because the PrivSpeech model size fits within the RPi4's processor cache, which has a 1MB LLC.

We also evaluate the impact of varying the top-k values on model performance. Reducing

Table 10: Comparison with prior work.

Work	Data	Obfuscation Method	Coverage	Model Size
Edgy [16]	Voice	Adversarial Learning	All Data	Constant
PriMask [72]	Mobile Sensors	Adversarial Learning	All Data	Constant
Olympus [120]	Image	Adversarial Learning	All Data	Constant
PrivSpeech	Voice	Static Adversarial Learning	Select Data	Scales top-K

k decreases the number of parameters, which reduces the model size, resulting in lower execution time, without notably affecting the privacy and utility scores. Furthermore, by quantizing the PrivSpeechNet from a 32-bit floating point to int8, we can further reduce the model size by 1/4 compared to storing fp32 representations, with only a marginal loss (not shown in the Table) of less than 1% in F_1 and Privacy scores.

Key Observations: Our analysis emphasizes the benefits of selectively targeting a reduced number of features for obfuscation, resulting in smaller model sizes and faster execution times. Specifically, we observed a significant reduction in model size by 1/3 when transitioning from top-240 to top-30 features. This highlights the efficiency gains achieved by prioritizing the most relevant features for obfuscation.

5.6 Related Work

Data minimization strategies, aims to reduce data collection and storage to mitigate the risk of personal information disclosure [94, 41, 43, 111, 30]. Simple methods to minimize data exposure rely on collecting the minimum necessary for the application. For example, voice assistants can use voice-activated wake-up commands to minimize data collection. However, as is the case for voice, even a minimal amount of data may contain private information along with useful data [130]. Privacy Preserving-inference solutions offer a more sophisticated approach by filtering the sensitive high-dimensional data with machine learning models [16, 52, 120, 131]. PrivSpeech belongs to this category.

Table 10 presents other works implementing data minimization techniques with privacy-preserving inference. The studies proposed in [16, 72, 120] use adversarial learning for training a neural network to obfuscate the data. These models are trained to minimize the loss against a utility model while maximizing the loss against adversary models. In contrast, PrivSpeech trains against specific target models in a subset of the data with a novel loss function that aims to minimize adversary losses to random guess predictions. Also, PrivSpeech is evaluated with interchanging privacies and utilities, showing the flexibility of the approach.

The work in [16] addresses the privacy concerns of voice user interfaces and digital assistants with EDGY, a lightweight framework that uses disentangled representation learning to filter sensitive information from voice data. Edgy uses a split network between the device and the server to filter sensitive data before sending it to the cloud. Unlike PrivSpeech, Edgy requires the service provider to host half of the inference model. In [72], PriMask is also a system that uses a neural network called MaskNet to obfuscate the data before transmission to reduce the adversary's ability to recover or extract specific personal attributes. PriMask relies on a Privacy Service Provider (PSP) for training and distribution. The system used a split adversarial learning method for generating new MaskNets and was evaluated in applications such as human activity recognition, urban environment crowdsensing, and driver behavior recognition. In comparison, PrivSpeech operates directly on the data without needing external services, only needing to specify the attacker models during training.

In [120], the authors present Olympus, a privacy framework that obfuscates sensor data to minimize personal information disclosure. Olympus is evaluated on Action recognition, Distracted Driving Detection, and Object Recognition as utility tasks and People Identity as privacy tasks. Differently PrivSpeech, Olympus uses adversarial training and also modifies all the data. PrivSpeech, however, can specifically target the top K features. Designed for voice data and evaluated on a constrained device.

PrivSpeech approaches the problem with a much more straightforward yet practical approach that can protect utility and privacy. Also, PrivSpeech offers improved robustness against specific adversaries since it is trained on specific static models with a lightweight model that does not need to learn to protect against varying adversaries and is easier to

evaluate since it only needs to train one model. In addition, PrivSpeech can also pre-emptily mitigate a re-trained adversary by removing the top k private features while restoring the utility and further obfuscating the remaining.

5.7 Discussion

UPE Modeling. In this Chapter, our focus was the evaluation of our mechanism for targeting specific sensitive features in human voice data to address $RQ3$. It is important to note that PrivSpeech is a privatizer as defined in Chapter 3. In this Chapter, we split the evaluation between Utility-Privacy tradeoffs and energy consumption since we focus on understanding the impact of targeting the top-k sensitive features and not comparing privatizers. However, PrivSpeech could be evaluated through the UPE framework presented in Chapter 3 where the top-k is studied as tuning parameters that could be further optimized along the UPE framework tradeoffs. Due to computational requirements for training PrivSpeechNet in every Top-K combination for every feature selection scheme, we focus on showing the trends and differences for set top-k values.

Energy Consumption. While our mechanism showed up to a 9% decrease in energy consumption when only targeting top-30 compared to the top-240 (i.e., targeting all features), we argue that other privacy-preserving solutions could benefit from our approach of targeting a subset of the features. The savings in energy consumption can be higher for privacy solutions that rely on algorithms of higher computing requirements. In the case of the human voice and the three particular tasks (i.e., speaker verification, gender, and emotion), we showed that a neural network model of 9-29 KB can offer privacy and maintain utility while being small enough for an Arduino UNO (ATmega328P Microcontroller) with 32KB of flash memory.

5.8 Conclusion

In this Chapter, we addressed $RQ3$ and developed a feature-driven privacy and utility-aware obfuscation mechanism for voice data. PrivSpeech can selectively obfuscate privacy-sensitive attributes, striking a delicate balance between data utility and privacy while offering different energy consumption tradeoffs depending on the size of the subset of sensitive data being obfuscated.

We also considered dynamic adversaries that can retrain the model on obfuscated data and showedthe relationship between important features in an attacker's inference model (linked to data privacy) and the user's requested services from the service provider (linked to data utility). This allows PrivSpeech to generate an obfuscation model that targets particular data segments considered more private to the user while preserving the rest.

6.0 Discussion

6.1 Utility, Privacy, and Energy Modeling

As discussed in Chapter 3, our approach is agnostic to the utility and private inferences models. In particular, we can incorporate any data transformation solution that protects against secondary data usage under a similar threat model (e.g., honest-but-curious service provider). Also, we note that the selection of a privatizer can be carried out offline. Thus, once a privatizer is selected, a reevaluation must only be performed if a new candidate privatizer is added or the user's and application's requirements change. The UPE framework assumes the availability of utility and privacy models capable of extracting sensitive information. However, if the models are not directly accessible, we can use proxy models for different utility and privacy tasks to understand the associated risks and benefits of using a privatizer.

Moreover, the evaluation of audio and image modalities case studies in Chapter 3 illustrate the selection process of privatizers and how to explore their hyperparameters as the energy consumption, utility, and privacy tradeoffs vary. Data transformations can affect utility and privacy tasks unpredictably, requiring a grid search to find optimal solutions. This happens because the relationship between the predictions of a model and the input data can be too complex, as is often the case for audio and image-based tasks that use deep neural networks. As a consequence, the privacy guarantees will be as good as the models and metrics chosen for the adversary's attacks.

In practice, the UPE optimization step should be done after the minimum objective thresholds are met, allowing the system designer or the user to weigh tradeoffs gains or prioritize objectives. Also, this type of evaluation can support the selection of privatizers in scenarios with dynamic requirements. For example, a camera may prioritize energy over privacy on specific periods or when a given condition is met (e.g., no person is in the image).

Finally, it is also important to consider that while deep neural networks could be retrained on privatized data for better performance, privatizers with lossy transformations can still

offer protection. For example, a blurred image may be recoverable. Previous research has shown that the amount of recoverable information will depend on the intensity of the blur filter [127]. For example, if the blur intensity is low, we can likely recover the obfuscated information [127]. However, increasing the intensity (e.g., the blurring effect) can destroy the utility of the data, as seen in Section 3.4.

6.2 Removing Sensitive Information

Removing sensitive information from data can be accomplished by adding noise or reducing data variability (e.g., blur filters). However, for resource-bound adversaries (static adversary), a simple data transformation such as min-max normalization may be enough to disrupt the performance of the adversary's method of extracting sensitive information. For example, suppose the adversary uses a large language model that requires thousands of hours to train. In that case, retraining the model to support a different data representation may be too expensive [31].

A **Static Adversary** is an attacker with fixed capabilities who attempts to compromise a system using fixed tools and methods. In the face of data transformations such as image blurring, such an adversary might struggle to derive meaningful insights.

When images are blurred, the discernible patterns within the data, which a model might have been trained to identify, are reduced or in the very least modified. For example, if a model has been trained to recognize faces, blurring those faces within the images can cause the model to fail.

From the model's perspective, the blurring transformation significantly increases the likelihood of incorrect predictions. This is because, even though the blurring process decreases the overall number of discernible patterns in the data, it nevertheless transforms the input space into a form quite different from the data on which the model was trained. The neural network's inherent complexity, coupled with its sensitivity to changes in input data, often leads to a deterioration in performance.

A **Dynamic Adversary**, in contrast, can update their tools and methods. A dynamic

adversary using neural network-based models can retrain to adapt his model to the transformed data. Consequently, the attacker can recover the model's performance if the privatizer does not effectively remove the patterns related to the sensitive tasks. This has been demonstrated in the results presented in Section 5.5.

Under such circumstances, mitigating the threat posed by a dynamic adversary calls for more drastic strategies that can incur utility loss as depicted in Section 5.5. This could include destructive information transformations such as noise addition or altogether removal of these features that can irreparably alter features or the data. These more robust measures ensure that the sensitive patterns cannot be discerned, regardless of how the adversary might retrain or adapt their models.

6.3 Energy Consumption of Privacy Solutions

The UPE model presented in Chapter 3 requires the empirical evaluation of the energy consumption for each device component in each utility and privacy model. While a time-consuming task, it can lead to sensing-cycle optimizations and a better understanding of the platform-specific energy consumption requirements. For example, Chapter 3 notes that energy could be saved by lowering the camera resolution combined with fewer blurring passes. The effort of the manual labor of this task can be mitigated by creating and sharing previously evaluated privatizers in an open-access database, allowing others to access information about previously evaluated privatizers.

Deep Learning Model Optimization. In recent years, significant advancements have been made in deep learning techniques to reduce models' size (e.g., pruning or quantization) [6, 69]. These techniques have successfully reduced the memory footprint of neural network models and enabled the development of lightweight models for constrained devices. In the context of privacy solutions, the work presented in Chapter 5 complements these techniques by demonstrating that the input size of an obfuscation model can be reduced to only target the private features, further reducing the total model size.

Interestingly, our findings presented in Section 5.5.6 show that, for the models we evalu-

ated, the utility and privacy remained unaffected by the lightweight model. This highlights the effectiveness of selectively targeting and modifying specific features while preserving overall model performance. Our work emphasizes the potential benefits of combining deep learning model compression techniques with feature selection for obfuscation purposes. Our results show that it is possible to achieve lightweight and efficient obfuscation models by leveraging both approaches without compromising utility or privacy.

6.4 Obfuscation with Neural Networks

In our study presented in Chapter 4 we also experimented with the Denoising autoencoder (DAE) proposed in [74]. We observed that the DAE neural network architecture was not as robust to noise addition as the sequence to point model, which was more amenable to the differential privacy framework. In particular, we added varying amounts of noise to the model parameter updates. For low values of noise (e.g., $z = 0.1$), the DAE model converged but required a high epsilon budget. However, we note that by increasing the number of clients participating in federated learning, we posit that it is possible to train the model with low ϵ values.

In Chapter 5, we pivot our approach and use a neural network to transform the data against an adversary using neural networks. We found that our approach is very effective depending on the adversary (static or dynamic) and the degree of overlap between the features shared by privacy and utility models. However, against static adversaries, PrivSpeechNet will learn how to transform the data such that it maximizes privacy and utility simultaneously. This happens because PrivSpeechNet is trained to learn to re-encode the data into a new representation that asserts utility inferences and forces wrong predictions for adversary models. In this setting, we demonstrate how achieving high levels of privacy and utility is possible, as substantiated by the results presented in Section 5.5.1.

Still, this approach will not be very effective against dynamic adversaries, which have been shown to regain previous performance after retraining. As a solution, the PrivSpeech framework employs a more rigorous approach by systematically removing sensitive features

while restoring utility in the data. In this context, we observe an inherent trade-off when sensitive features overlap with utility features. If both utility and privacy tasks depend on the same shared features, restoring utility will decrease privacy. This phenomenon refines the classic utility-privacy Pareto frontier trade-off to the level of feature importance of the data.

Hence, if the data needed by the utility and privacy tasks do not share the same underlying information, PrivSpeech can effectively hinder the inference of a retrained adversary while retaining the utility of the data. In the presence of static adversaries, PrivSpeech can successfully modify the data to create representations that preserve both utility and privacy, regardless of the interdependence of features.

Explainable AI. In Chapter 5, our work highlights the crucial step of selectively identifying sensitive features, which plays a vital role in balancing utility and privacy, particularly in scenarios where distinguishing essential features for both aspects is challenging. While our technique leverages SHAP (SHapley Additive exPlanations) and other algorithms to aid in feature selection, it also emphasizes the significance of transparency in artificial intelligence (AI) to identify and understand sensitive features. The field of explainable AI is still in its early stages. However, further advancements in understanding the importance of features for a given model can significantly benefit PrivSpeech and similar approaches. We can refine our obfuscation strategies and enhance the system's overall effectiveness by gaining insights into which features influence utility and privacy.

6.5 Data Secondary Use and Differential Privacy

Secondary data usage risk stems from data collected for one purpose (primary use) that can be repurposed or reused for other unintended purposes (secondary use). For instance, data collected by smart thermostats can have the *primary use* as informing the house's average temperature while an attacker can attempt to infer the users' behaviors, preferences, or habits as *secondary use* of the data.

Differential privacy is a robust privacy-preserving technique that guarantees a certain

level of privacy by adding a controlled amount of noise to the data or the analysis output. The core idea is to ensure that removing or adding a single data item does not significantly change the probability distribution of the query outputs, hence hiding any individual data item. As shown in Chapter 4, differential privacy can effectively anonymize each entry's existence in relation to some query (e.g., the resulting weights of a trained model or the model's inferences) in the dataset.

Finally, we note that the proposed approaches from Chapter 3 and Chapter 5 and differential privacy can complement each other when the IoT sensor data relates to multiple users. Our PrivSpeechNet approach could be combined with differential privacy to provide additional privacy protection by anonymizing any specific dataset entry. For instance, one could first apply differential privacy techniques when collecting and storing data to protect individual records, then apply our method when building and deploying predictive models further to protect privacy at the level of model predictions.

7.0 Conclusion

The volume of data generated by the IoT brings opportunities and difficulties for both users and attackers. IoT devices can offer many features and sensing capabilities that generate more data than applications need. The removal of private content from shared data has to be done at the source of data collection to reduce exposure. Hence, energy-constrained IoT devices will need to optimize for application performance, privacy guarantees, and energy consumption of privatizers.

This dissertation demonstrated the systematic selection of optimal privatizers through the lens of Utility, Privacy, and Energy (UPE) trade-offs for IoT applications. By conducting comprehensive evaluations on image and audio data, we demonstrated our model's applicability across two distinct modalities, finding privatizers that balance energy efficiency and data utility.

We further explored data privacy protection in a federated learning setting, developing a framework to evaluate differential privacy measures. In this context, we focused on using non-intrusive load monitoring techniques to infer appliance usage from comprehensive household energy consumption data. We learned that differential privacy guarantees and the impact on the utility of the data can vary significantly depending on the model being trained with federated learning.

The culmination of this research is reflected in PrivSpeech - a specialized privatizer for human voice data. PrivSpeech meticulously targets specific sensitive features for obfuscation while maintaining the integrity of the remaining data. Its effectiveness is demonstrated across three different datasets and against static and dynamic adversaries (those who can or cannot retrain their models on obfuscated data). We learned that it is possible to create smaller, energy-efficient, lightweight obfuscation models by targeting specific features in the dataset while retaining privacy and utility.

Through the journey of this dissertation, we have made substantial progress toward formulating energy-aware and utility-preserving privacy solutions. These solutions aim to protect users' privacy while minimizing changes in the data. Our successful development of a

framework to characterize privatizers for protection against secondary data usage adversaries presents a significant leap toward enhancing IoT systems. It assures a more robust privacy protection framework, carefully balanced with preserving data utility and considering energy constraints typical of IoT deployments.

7.1 Contributions

Through this dissertation, we contribute to the knowledge body on utility awareness, privacy preservation, and energy-efficient privacy solutions for resource-constrained devices. We also highlight how each contribution relates to the challenges introduced in Section 1.2.1; in particular, we make the following contributions:

1. **Design, implementation, and evaluation of the UPE Model**: We developed a novel model to harmonize Utility, Privacy, and Energy (UPE) considerations in IoT systems. Our model directs users to viable strategies for reducing energy consumption while maintaining a robust balance of privacy and utility. We have implemented and assessed our model with image and audio tasks and different privatizers, demonstrating the successful identification and application of efficient 'privatizers' for each task. Finally, we show an in-depth analysis of UPE tradeoffs that reveals their nonlinear characteristics as hyperparameters of privatizers vary. This analysis emphasizes the complexity of selecting optimal privatizers.
    - **Evaluation Methodology of Privatizers**: A comprehensive evaluation of image and audio privatizers across two case studies with the proposed UPE model. This evaluation aids in understanding the utility-privacy tradeoffs while distinguishing energy-efficient privatizers. The proposed model enabled the selection of better privatizers by identifying candidates with similar UP tradeoffs but less energy consumption for the image classification task. In the Audio modality, we designed a simple privatizer based on image blur kernels that showed the best performance along the UPE tradeoffs. Moreover, we highlight that the leading cause of the energy

consumption of privacy solutions is not the computation complexity of the algorithm but the time taken to apply the obfuscation.

This contribution assesses Challenge 1 since we developed a framework that characterizes privatizers' UPE tradeoffs. Challenges 2, 4, 5 are hinted at with the FaceBlur privatizer, where we minimize the impact on the utility (utility-preserving) of the data by focusing the privatizer in the facial region (sensitive feature consideration).

2. **Development and evaluation of the DPFL Framework**: The creation of a differentially-private federated learning (DPFL) framework to train Non-Intrusive Load Monitoring (NILM) models. This framework offers a privacy-preserving distributed learning system that effectively mitigates privacy attacks. The DPFL framework was implemented as open-source modules, with integration capabilities for privacy attacks to measure DPFL's effectiveness in preserving privacy. This includes developing interfaces that allow extensions to existing NILM models and datasets.
   - **Evaluation Methodology of NILM Neural Network Models**: An evaluation of the different NILM models within the DPFL framework, providing insights into how different models behave with DP noise. Our findings can be used to develop neural-network models that are more resilient to DP noise.

   Here, we assess privacy-preserving solutions in the context of federated learning and differential privacy as a privatizer. In particular, we give a solution to challenge 3 (Privacy-Preserving Machine Learning) for non-intrusive load monitoring applications that use neural networks.
3. **Design of PrivSpeechNet Obfuscation**: The design of *PrivSpeech*, an obfuscation mechanism for voice utility and privacy tasks that strategically identifies and obfuscates sensitive features while preserving the integrity of the remaining features towards providing high utility for users.
   - **Exploration Methodology of Feature Selection Strategies**: Exploring different top-k feature selection strategies to inform task-specific voice obfuscation algorithms. This exploration leverages Shapley values to efficiently erase sensitive features from datasets, optimizing the privacy-utility balance.

PrivSpeech tackles Challenge 2 by showing a generic approach for targeting the sensitive features of data while minimizing the impact on utility (Challenge 2 and 4). PrivSpeech also shows how neural-network can be trained fool adversary models to predict no better than random, which answers challenge 3 (Privacy-Preserving Machine Learning). Finally, we also address challenge 5 by creating a model that could be quantized to int8 without losing performance.

The results of this research and methodology for evaluation, for example, from the UPE model, can serve as the foundation for other researchers and practitioners to understand essential aspects in designing future privacy solutions for constrained devices.

7.2 Summary

This dissertation provides compelling evidence to support our central thesis as presented in Section 1.2:

> *"It is possible to design privacy tools specifically for Internet of Things (IoT) based resource-constrained devices, emphasizing mechanisms that enable local processing to achieve a delicate balance between privacy protection, utility preservation, and energy consumption."*

In addressing the first research question, "How can the characteristics of Utility, Privacy, and Energy (UPE) tradeoffs be understood and evaluated within the context of privacy-preserving functions in IoT devices?" - this dissertation has demonstrated through contribution 1 the ability to systematically select optimal privatizers that balance the UPE trade-offs for IoT applications. The developed model's applicability to different data modalities, such as image and audio, provides a broad foundation for exploring software-based privacy solutions for IoT.

The second research question - "What methods can be formulated to maintain both privacy and utility of smart meter data from IoT devices, specifically in a setting that utilizes federated learning" - is affirmed by contribution 2 that design, implements, and evaluates the use of differential privacy to protect user's participation in the outputs of a neural net model trained through federated learning for non-intrusive load monitoring. We showed different

privacy guarantees measured with attackers' advantage metric for different epsilon values. We also noted that neural network architecture can be more resilient to the noise added by differential privacy.

Regarding the third research question - "How to selectively obfuscate sensitive data features while keeping utility information intact and minimize the computing requirements for constrained resources?" - again, contributions (1-3) show our UPE framework and PrivSpeech to contrast how the selection of features and can impact the UPE tradeoffs. In particular, contribution 3 show the different tradeoffs depending on the attacker's capability of retraining the model and how the utility and privacy co-dependence of features can affect the obfuscation mechanism.

Finally, PrivSpeech can be extrapolated for multiple users with multiple utilities and privacies requirements. Users can maximize the utilities and privacy against static adversaries. However, they will have to choose tradeoffs against dynamic adversaries based on the extent that utility features are codependent with sensitive information features.

This dissertation validates the thesis statement and comprehensively responds to each research question. It establishes that privacy-preserving functions, or privatizers, can be successfully applied in IoT systems, factoring in Utility, Privacy, and Energy (UPE) trade-offs, without compromising user privacy and data utility, even in energy-constrained environments.

7.3 Future Work

Next, we discuss some directions for future work related to this research.

7.3.1 Extending the Utility, Privacy, and Energy (UPE) Model to Other Data Modalities

The UPE model in this dissertation was demonstrated with image and audio data. One promising direction of future research would be to extend the model to other data modalities

such as textual, geospatial, or multimodal data. These may present different challenges and require the development of novel privatizers to protect user privacy effectively while maintaining data utility and considering energy efficiency.

7.3.2 Multi-modal Privacy Protection

With the widespread adoption of IoT devices, data is often collected in multiple modalities. For example, in a smart home context, data could be collected from cameras (visual), microphones (audio), temperature sensors (thermal), and many more. An advanced adversary could exploit these multiple data sources, cross-referencing information to infer sensitive details that might not be extractable from a single data source.

Privacy protection in such multi-modal scenarios presents a significant challenge. A straightforward application of independent privacy-preserving functions on each data modality might not suffice. The adversary could still cross-reference the obfuscated data from different modalities to infer sensitive information. Thus, privacy protection in this scenario necessitates a comprehensive understanding of the correlations between different data modalities and advanced privacy-preserving functions that can account for these correlations.

Future work in this area could explore the development of such advanced multi-modal privacy-preserving functions. These functions would need to be able to process and obfuscate data from different modalities in a coordinated manner, ensuring that cross-referencing obfuscated data does not compromise privacy. Furthermore, these functions would also need to consider the energy constraints typical of IoT devices, ensuring that multi-modal privacy protection does not compromise the usability of IoT systems.

This research direction presents exciting opportunities for advancing the field of privacy in IoT and could pave the way for developing highly secure multi-modal IoT systems. It would help protect users' privacy and enhance trust in IoT technologies, accelerating their adoption in various sectors.

7.3.3 Privatizer Adaptation in Dynamic IoT Environments

The IoT environment is characterized by its high dynamism. Devices often enter or exit the network, users might change their privacy preferences, energy availability can vary, or new tasks may be required. This ever-changing scenario can pose significant challenges to maintaining privacy. Future work could explore the development of adaptable privatizers that can adjust to these changes in real-time. For instance, if a new device enters the network, the privatizer might need to adjust to incorporate data from this device without compromising privacy. If the energy source changes - from a constant power supply to a battery-operated one - the privatizer might need to adapt to be more energy-efficient. Developing such adaptable privatizers would require a deep understanding of the dynamicity of IoT environments and advanced AI techniques to allow for real-time adaptation.

7.3.4 Privacy Protection for Interconnected IoT Systems

The current IoT landscape consists of numerous interconnected devices that often share and process data jointly. This interconnectedness can make privacy protection more challenging, as privacy breaches in one device could affect the entire system. Moreover, these interconnected systems might have multiple layers with varying sensitivity levels. For instance, a smart home system might contain:

- a security system layer that requires a high degree of privacy,
- a utility layer with a lower sensitivity level, and
- a leisure layer where privacy is not critical.

Future work could explore developing multi-level privacy solutions that can provide an appropriate level for each layer and ensure privacy across the interconnected system. This would involve understanding the interaction between IoT systems and the data flow within and between these systems and developing advanced privacy-preserving functions to manage these complex scenarios.

Bibliography

[1] Emo-DB, January 2013. [Online; accessed 28. Jun. 2023].

[2] Resources | NIST, April 2020. [Online; accessed 2. Jul. 2023].

[3] COCO - Common Objects in Context, Aug 2021. [Online; accessed 12. Oct. 2021].

[4] Module: tff | TensorFlow Federated, Feb 2021. [Online; accessed 18. Feb. 2021].

[5] TIMIT Acoustic-Phonetic Continuous Speech Corpus - Linguistic Data Consortium, Oct 2021. [Online; accessed 12. Oct. 2021].

[6] TensorFlow Lite, May 2022. [Online; accessed 19. Jun. 2023].

[7] SmartThings Developers, July 2023. [Online; accessed 2. Jul. 2023].

[8] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Oct 2016.

[9] David Ahmedt-Aristizabal, Clinton Fookes, Sasha Dionisio, Kien Nguyen, Joao Paulo S Cunha, and Sridha Sridharan. Automated analysis of seizure semiology and brain electrical activity in presurgery evaluation of epilepsy: A focused survey. *Epilepsia*, 58(11):1817–1831, 2017.

[10] David Ahmedt-Aristizabal, Clinton Fookes, Kien Nguyen, Simon Denman, Sridha Sridharan, and Sasha Dionisio. Deep facial analysis: A new phase i epilepsy evaluation using computer vision. *Epilepsy & Behavior*, 82:17–24, 2018.

[11] David Ahmedt-Aristizabal, Kien Nguyen, Simon Denman, Sridha Sridharan, Sasha Dionisio, and Clinton Fookes. Deep motion analysis for epileptic seizure classification. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3578–3581. IEEE, 2018.

[12] Karim Alghoul, Saeed Alharthi, Hussein Al Osman, and Abdulmotaleb El Saddik. Heart Rate Variability Extraction From Videos Signals: ICA vs. EVM Comparison. *IEEE Access*, 5:4711–4719, Mar 2017.

[13] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. *arXiv*, Jul 2017.

[14] Md Zahangir Alom, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S. Awwal, and Vijayan K. Asari. The history began from alexnet: A comprehensive survey on deep learning approaches, 2018.

[15] Ranya Aloufi, Hamed Haddadi, and David Boyle. Edgy: On-device paralinguistic privacy protection. In *Proc. of the 12th ACM Wireless of (S3) Workshop*. ACM, 2021.

[16] Ranya Aloufi, Hamed Haddadi, and David Boyle. Paralinguistic Privacy Protection at the Edge. *ACM Trans. Priv. Secur.*, 26(2):1–27, April 2023.

[17] Prabhanjan Ananth, Aayush Jain, Huijia Lin, Christian Matt, and Amit Sahai. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 284–332, Cham, 2019. Springer International Publishing.

[18] Devansh Arpit, Stanislaw Jastrz kebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 233–242. PMLR, 06–11 Aug 2017.

[19] M. R. Asghar, G. Dán, D. Miorandi, and I. Chlamtac. Smart meter data privacy: A survey. *IEEE Communications Surveys Tutorials*, 19(4):2820–2835, Fourthquarter 2017.

[20] Brendan Avent, Javier Gonzalez, Tom Diethe, Andrei Paleyes, and Borja Balle. Automatic discovery of privacy–utility pareto fronts. *Proceedings on Privacy Enhancing Technologies*, 2020(4):5–23, 2020.

[21] Saurabh Bagchi, Tarek F. Abdelzaher, Ramesh Govindan, Prashant Shenoy, Akanksha Atrey, Pradipta Ghosh, and Ran Xu. New frontiers in iot: Networking, systems, reliability, and security challenges. *IEEE Internet of Things Journal*, 7(12):11330–11346, 2020.

[22] David E Bakken, R Rarameswaran, Douglas M Blough, Andy A Franz, and Ty J Palmer. Data obfuscation: Anonymity and desensitization of usable data sets. *IEEE Security & Privacy*, 2(6):34–41, 2004.

[23] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *arXiv preprint arXiv:1807.01647*, 2018.

[24] Borja Balle, James Bell, Adria Gascon, and Kobbi Nissim. The Privacy Blanket of the Shuffle Model. *arXiv*, Mar 2019.

[25] Karim Said Barsim and Bin Yang. On the feasibility of generic deep disaggregation for single-load extraction. *CoRR*, abs/1802.02139, 2018.

[26] Nipun Batra, Rithwik Kukunuri, Ayush Pandey, Raktim Malakar, Rajat Kumar, Odysseas Krystalakos, Mingjun Zhong, Paulo Meira, and Oliver Parson. Towards reproducible state-of-the-art energy disaggregation. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '19, page 193–202, New York, NY, USA, 2019. Association for Computing Machinery.

[27] Rachel Bittner. Python wrapper around sox, Mar 2022. [Online; accessed 6. Mar. 2022].

[28] Kaitlin Boeckl, Michael Fagan, William Fisher, Naomi Lefkovitz, Katerina Megas, Ellen Nadeau, Ben Piccarreta, Danna Gabel O'Rourke, and Karen Scarfone. Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks. *CSRC | NIST*, Jun 2019.

[29] John Borking, P. Verhaar, B.M.A. Eck, P. Siepel, G.W. Blarkom, R. Coolen, M. Uyl, J. Holleman, P. Bison, R. Veer, J. Giezen, Andrew Patrick, C. Holmes, J.C.A. Lubbe, Roy Lachman, S. Kenny, Randy Song, K. Cartrysse, J. Huizenga, and X. Zhou. *Handbook of Privacy and Privacy-Enhancing Technologies The case of Intelligent Software Agents*. 11 2003.

[30] Phuthipong Bovornkeeratiroj, Srinivasan Iyengar, Stephen Lee, David Irwin, and Prashant Shenoy. RepEL: A Utility-Preserving Privacy System for IoT-Based Energy Meters. IEEE, 2020.

[31] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv*, May 2020.

[32] A.J. Bernheim Brush, Bongshin Lee, Ratul Mahajan, Sharad Agarwal, Stefan Saroiu, and Colin Dixon. Home automation in the wild: Challenges and opportunities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 2115–2124, New York, NY, USA, 2011. Association for Computing Machinery.

[33] Hui Cao, Shubo Liu, Renfang Zhao, and Xingxing Xiong. IFed: A novel federated learning framework for local differential privacy in Power Internet of Things. *Int. J. Distrib. Sens. Netw.*, 16(5):1550147720919698, May 2020.

[34] Ajesh Koyatan Chathoth, Abhyuday Jagannatha, and Stephen Lee. Federated intrusion detection for iot with heterogeneous cohort privacy. *arXiv preprint arXiv:2101.09878*, 2021.

[35] Jung Hee Cheon, Wonhee Cho, Minki Hhan, Jiseung Kim, and Changmin Lee. Statistical zeroizing attack: Cryptanalysis of candidates of bp obfuscation over ggh15 multilinear map. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 253–283, Cham, 2019. Springer International Publishing.

[36] Sherman S. M. Chow. Can we securely outsource big data analytics with lightweight cryptography? In *Proceedings of the Seventh International Workshop on Security in Cloud Computing*, SCC '19, pages 1–1, New York, NY, USA, 2019. ACM.

[37] Giovanni Costantini, Iacopo Iaderola, Andrea Paoloni, and Massimiliano Todisco. EMOVO corpus: an Italian emotional speech database. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3501–3504, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

[38] Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham Mysore, Fredo Durand, and William T. Freeman. The visual microphone: Passive recovery of sound from video. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 33(4):79:1–79:10, 2014.

[39] Abe Davis, Michael Rubinstein, Neal Wadhwa, Gautham J. Mysore, Frédo Durand, and William T. Freeman. The visual microphone: Passive recovery of sound from video. *ACM Trans. Graph.*, 33(4), July 2014.

[40] Defconconference. DEF CON Safe Mode - Paul Marrapese - Abusing P2P to Hack 3 Million Cameras, Aug 2020. [Online; accessed 7. Sep. 2020].

[41] Mehmet Özgün Demir, Güneş Karabulut Kurt, Volker Lücken, Gerd Ascheid, and Guido Dartmann. Impact of the communication channel on information theoretical privacy. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 2870–2874. IEEE, June 2017.

[42] Wenxiu Ding, Xuyang Jing, Zheng Yan, and Laurence T. Yang. A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion. *Information Fusion*, 51:129 – 144, 2019.

[43] Roy Dong, Lillian J. Ratliff, Alvaro A. Cárdenas, Henrik Ohlsson, and S. Shankar Sastry. Quantifying the Utility–Privacy Tradeoff in the Internet of Things. *ACM Trans. Cyber-Phys. Syst.*, 2(2):1–28, Jun 2018.

[44] Jennifer B. Dunn, Linda Gaines, Jarod C. Kelly, and Kevin G. Gallagher. *Life Cycle Analysis Summary for Automotive Lithium-Ion Battery Production and Recycling*, pages 73–79. Springer International Publishing, Cham, 2016.

[45] Cynthia Dwork. Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340, 2011.

[46] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[47] M. El Soussi, P. Zand, F. Pasveer, and G. Dolmans. Evaluating the performance of emtc and nb-iot for smart city applications. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2018.

[48] Samy El-Tawab, Raymond Oram, Michael Garcia, Chris Johns, and B. Brian Park. Data analysis of transit systems using low-cost IoT technology. *2017 IEEE Interna-*

tional Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pages 497–502, Mar 2017.

[49] Ferdinando Fioretto, Terrence W. K. Mak, and Pascal Van Hentenryck. Privacy-preserving obfuscation of critical infrastructure networks. *CoRR*, abs/1905.09778, 2019.

[50] Noria Foukia, David Billard, and Eduardo Solana. PISCES: A framework for privacy by design in IoT. *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pages 706–713, Dec 2016.

[51] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015.

[52] Yongjian Fu, Shuning Wang, Linghui Zhong, Lili Chen, Ju Ren, and Yaoxue Zhang. Svoice: Enabling voice communication in silence via acoustic sensing on commodity devices. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, SenSys '22, page 622–636, New York, NY, USA, 2023. Association for Computing Machinery.

[53] O. Garcia-Morchon, S. Kumar, and M. Sethi. Internet of things (iot) security: State of the art and challenges. RFC 8576, RFC Editor, April 2019.

[54] Google. Tensorflow detection model zoo, Oct 2019. [Online; accessed 22. Oct. 2019].

[55] Emma Graham-Harrison and Carole Cadwalladr. Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. *the Guardian*, Mar 2018.

[56] Yunzhe Guo, Dan Wang, Arun Vishwanath, Cheng Xu, and Qi Li. Towards federated learning for hvac analytics: A measurement study. In *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, pages 68–73, 2020.

[57] William Haack, Madeleine Severance, Michael Wallace, and Jeremy Wohlwend. Security analysis of the amazon echo. *Allen Institute for Artificial Intelligence*, 2017.

[58] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep speech: Scaling up end-to-end speech recognition, 2014.

[59] Alon Harell, Stephen Makonin, and Ivan V Bajić. Wavenilm: A causal neural network for power disaggregation from the complex power signal. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8335–8339. IEEE, 2019.

[60] Todd Haselton. Google admits partners leaked more than 1,000 private conversations with Google Assistant. *CNBC*, Jul 2019.

[61] Zaobo He, Zhipeng Cai, Yunchuan Sun, Yingshu Li, and Xiuzhen Cheng. Customized privacy preserving for inherent data and latent data. *Pers. Ubiquit. Comput.*, 21(1):43–54, February 2017.

[62] Julia Hirschberg, Anna Hjalmarsson, and Noémie Elhadad. You're as sick as you sound: Using computational approaches for modeling speaker state to gauge illness and recovery.

[63] Ming Hua and Jian Pei. A survey of utility-based privacy-preserving data transformation methods. In *Privacy-Preserving Data Mining*, 2008.

[64] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. Generative Adversarial Privacy. *arXiv*, Jul 2018.

[65] Chong Huang, Peter Kairouz, Xiao Chen, Lalitha Sankar, and Ram Rajagopal. Generative Adversarial Privacy. *arXiv*, Jul 2018.

[66] Alexander Hubers, Emily Andrulis, Levi Scott, Tanner Stirrat, Duc Tran, Ruonan Zhang, Ross Sowell, Cindy Grimm, and William D. Smart. Video manipulation techniques for the protection of privacy in remote presence systems, 2015.

[67] Apple Inc. Apple Home - Apple Developer, July 2023. [Online; accessed 2. Jul. 2023].

[68] David Ingram. Facebook says data leak hits 87 million users, widening privacy scandal, Oct 2019. [Online; accessed 4. Oct. 2019].

[69] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[70] Bargav Jayaraman and David Evans. Evaluating Differentially Private Machine Learning in Practice, 2019. [Online; accessed 18. Feb. 2021].

[71] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. Revisiting membership inference under realistic assumptions, 2021.

[72] Linshan Jiang, Qun Song, Rui Tan, and Mo Li. PriMask: Cascadable and Collusion-Resilient Data Masking for Mobile Cloud Inference. In *SenSys '22: Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 164–178. Association for Computing Machinery, New York, NY, USA, November 2022.

[73] Samira Ebrahimi Kahou, Xavier Bouthillier, Pascal Lamblin, Caglar Gulcehre, Vincent Michalski, Kishore Konda, Sébastien Jean, Pierre Froumenty, Yann Dauphin, Nicolas Boulanger-Lewandowski, et al. Emonets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 10(2):99–111, 2016.

[74] Jack Kelly and William Knottenbelt. Neural nilm: Deep neural networks applied to energy disaggregation. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, BuildSys '15, page 55–64, New York, NY, USA, 2015. Association for Computing Machinery.

[75] Jack Kelly and William Knottenbelt. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. 2(150007), 2015.

[76] Bogil Kim, Sungjae Lee, Amit Ranjan Trivedi, and William J. Song. Energy-Efficient Acceleration of Deep Neural Networks on Realtime-Constrained Embedded Edge Devices. *IEEE Access*, Nov 2020.

[77] Dae Hyun Kim, Taeyoung Kong, and Seungbin Jeong. Finding Solutions to Generative Adversarial Privacy. *arXiv*, Oct 2018.

[78] J Zico Kolter and Matthew J Johnson. Redd: A public data set for energy disaggregation research. In *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*, volume 25, pages 59–62, 2011.

[79] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International*

Conference on Neural Information Processing Systems - Volume 1, NIPS'12, page 1097–1105, Red Hook, NY, USA, 2012. Curran Associates Inc.

[80] Nicholas D. Lane, Sourav Bhattacharya, Akhil Mathur, Petko Georgiev, Claudio Forlivesi, and Fahim Kawsar. Squeezing Deep Learning into Mobile and Embedded Devices. *IEEE Pervasive Comput.*, 2017.

[81] Eric C. Larson, TienJui Lee, Sean Liu, Margaret Rosenfeld, and Shwetak N. Patel. Accurate and privacy preserving cough sensing using a low-cost microphone. In *UbiComp '11*. ACM, 2011.

[82] M. Lauridsen, I. Z. Kovacs, P. Mogensen, M. Sorensen, and S. Holst. Coverage and capacity analysis of lte-m and nb-iot in a rural area. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, Sep. 2016.

[83] Amanda Lazar, Christian Koehler, Joshua Tanenbaum, and David H Nguyen. Why we use and abandon smart devices. In *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, pages 635–646, 2015.

[84] K. Lebart, J.-M. Boucher, and P. Denbigh. A new method based on spectral subtraction for speech dereverberation. *Acta Acustica united with Acustica*, 87, 2001.

[85] June-Goo Lee, Sanghoon Jun, Young-Won Cho, Hyunna Lee, Guk Bae Kim, Joon Beom Seo, and Namkug Kim. Deep Learning in Medical Imaging: General Overview. *Korean Journal of Radiology*, 18(4):570–584, Aug 2017.

[86] Wei Li, Flávia C. Delicato, and Albert Y. Zomaya. Adaptive energy-efficient scheduling for hierarchical wireless sensor networks. *ACM Trans. Sen. Netw.*, 9(3), June 2013.

[87] Xinyu Li, Yanyi Zhang, Ivan Marsic, Aleksandra Sarcevic, and Randall S Burd. Deep learning for rfid-based activity recognition. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 164–175. ACM, 2016.

[88] Bingyan Liu, Yuanchun Li, Yunxin Liu, Yao Guo, and Xiangqun Chen. PMC: A Privacy-preserving Deep Learning Model Customization Framework for Edge Computing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(4):1–25, December 2020.

[89] Jingwen Liu, Yanlei Gu, and Shunsuke Kamijo. Joint customer pose and orientation estimation using deep neural network from surveillance camera. In *2016 IEEE International Symposium on Multimedia (ISM)*, pages 216–221. IEEE, 2016.

[90] Steven R. Livingstone and Frank A. Russo. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), April 2018. Funding Information Natural Sciences and Engineering Research Council of Canada: 2012-341583 Hear the world research chair in music and emotional speech from Phonak.

[91] Gustavo López, Luis Quesada, and Luis A Guerrero. Alexa vs. siri vs. cortana vs. google assistant: a comparison of speech-based natural user interfaces. In *International Conference on Applied Human Factors and Ergonomics*, pages 241–250. Springer, 2017.

[92] David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, and Aaron Lefohn. Gpgpu: General purpose computation on graphics hardware. In *ACM SIGGRAPH 2004 Course Notes*, SIGGRAPH '04, page 33–es, New York, NY, USA, 2004. Association for Computing Machinery.

[93] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.

[94] Mohammad Malekzadeh, Richard G. Clegg, Andrea Cavallaro, and Hamed Haddadi. Privacy and Utility Preserving Sensor-Data Transformations. *arXiv*, Nov 2019.

[95] H. Brendan McMahan and Galen Andrew. A general approach to adding differential privacy to iterative training procedures. *CoRR*, abs/1812.06210, 2018.

[96] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data, 2017.

[97] Richard McPherson, Reza Shokri, and Vitaly Shmatikov. Defeating image obfuscation with deep learning. *CoRR*, abs/1609.00408, 2016.

[98] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, volume 7, pages 94–103, 2007.

[99] Andrés Molina-Markham, Prashant Shenoy, Kevin Fu, Emmanuel Cecchet, and David Irwin. Private memoirs of a smart meter. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, BuildSys '10, page 61–66, New York, NY, USA, 2010. Association for Computing Machinery.

[100] Daniel Mossé, Henrique Pötter, and Stephen Lee. Maintaining privacy and utility in iot system analytics. In *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 157–164. IEEE, 2020.

[101] Daniel Mossé, Henrique Pötter, and Stephen Lee. Maintaining Privacy and Utility in IoT System Analytics. In *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 157–164. IEEE, October 2020.

[102] D. Mossé, H. Pötter, and S. Lee. Maintaining privacy and utility in iot system analytics. In *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 157–164, 2020.

[103] Daniel Mossé, Henrique Pötter, and Stephen Lee. Maintaining privacy and utility in iot system analytics. In *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, pages 157–164, 2020.

[104] Alison Noble. *Protecting privacy in practice: The current use, development and limits of Privacy Enhancing Technologies in data analysis*. Royal Society, 2019.

[105] Witold Oleszkiewicz, Peter Kairouz, Karol Piczak, Ram Rajagopal, and Tomasz Trzciński. Siamese Generative Adversarial Privatizer for Biometric Data. *SpringerLink*, pages 482–497, Dec 2018.

[106] S.R.M. Oliveira and O.R. Zaiane. Protecting sensitive knowledge by data sanitization. In *3rd IEEE Int. Conference on Data Mining*, 2003.

[107] opencv. opencv_contrib, Sep 2021. [Online; accessed 21. Sep. 2021].

[108] Francisco Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.

[109] Tribhuvanesh Orekondy, Mario Fritz, and Bernt Schiele. Connecting pixels to privacy and utility: Automatic redaction of private information in images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8466–8475, 2018.

[110] Pankesh Patel, Muhammad Intizar Ali, and Amit Sheth. On using the intelligent edge for iot analytics. *IEEE Intelligent Systems*, 32(5):64–69, 2017.

[111] Charith Perera, Mahmoud Barhamgi, Arosha K. Bandara, Muhammad Ajmal, Blaine Price, and Bashar Nuseibeh. Designing Privacy-aware Internet of Things Applications. *arXiv*, Mar 2017.

[112] Charith Perera, Ciaran McCormick, Arosha K. Bandara, Blaine A. Price, and Bashar Nuseibeh. Privacy-by-design framework for assessing internet of things applications and platforms. In *Proceedings of the 6th International Conference on the Internet of Things*, IoT'16, page 83–92, New York, NY, USA, 2016. Association for Computing Machinery.

[113] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials*, 16(1):414–454, First 2014.

[114] Lionel Pigou, Aäron Van Den Oord, Sander Dieleman, Mieke Van Herreweghe, and Joni Dambre. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *International Journal of Computer Vision*, 126(2-4):430–439, 2018.

[115] Henrique Pötter, Stephen Lee, and Daniel Mossé. Towards Privacy-preserving Framework for Non-Intrusive Load Monitoring. In *e-Energy '21: Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, pages 259–263. Association for Computing Machinery, New York, NY, USA, June 2021.

[116] Henrique Pötter, Daniel Mossé, and Stephen Lee. Bringing Energy into Utility-Privacy Tradeoff in IoT. In *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 116–123. IEEE, June 2022.

[117] Henrique Brittes Pötter and Alexandre Sztajnberg. Adapting heterogeneous devices into an iot context-aware infrastructure. In *Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '16, page 64–74, New York, NY, USA, 2016. Association for Computing Machinery.

[118] rabitt. pysox, Sep 2021. [Online; accessed 21. Sep. 2021].

[119] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Swish: a Self-Gated Activation Function. *arXiv: Neural and Evolutionary Computing*, 2017.

[120] Nisarg Raval, Ashwin Machanavajjhala, and Jerry Pan. Olympus: Sensor privacy through utility aware obfuscation. *Proceedings on Privacy Enhancing Technologies*, 2019(1), 2019.

[121] Mirco Ravanelli and Yoshua Bengio. Speaker recognition from raw waveform with sincnet, 2019.

[122] Ju Ren, Hui Guo, Chugui Xu, and Yaoxue Zhang. Serving at the edge: A scalable iot architecture based on transparent computing. *IEEE Network*, 31(5):96–105, 2017.

[123] Vagner Sacramento, Markus Endler, and Clarisse Souza. A privacy service for location-based collaboration among mobile users. *Journal of the Brazilian Computer Society*, 14(4):41–57, 2008.

[124] Mukesh K. Saini, Pradeep K. Atrey, Sharad Mehrotra, and Mohan S. Kankanhalli. Privacy aware publication of surveillance video. *International Journal of Trust Management in Computing and Communications*, Mar 2013.

[125] Jonathan Schnader. Alexa, are you a foreign agent: Confronting the risk of foreign intelligence exploitation of private home networks, home assistants, and connectivity in the security clearance process. *Rich. JL & Tech.*, 25:1, 2018.

[126] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. *arXiv*, Mar 2015.

[127] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graphics*, 2008.

[128] M Sharifa, Roland Goecke, Michael Wagner, Julien Epps, Michael Breakspear, Gordon Parker, et al. From joyous to clinically depressed: Mood detection using spontaneous speech. In *Twenty-Fifth International FLAIRS Conference*, 2012.

[129] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[130] Rita Singh. *Profiling Humans from their Voice*. Springer, Singapore, 2019.

[131] Brij Srivastava, Aurélien Bellet, Marc Tommasi, and Emmanuel Vincent. Privacy-preserving adversarial representation learning in asr: Reality or illusion? pages 3700–3704, 09 2019.

[132] Nick Statt. Amazon sent 1,700 Alexa voice recordings to the wrong user following data request. *Verge*, Dec 2018.

[133] Yoshihiko Suhara, Yinzhan Xu, and Alex'Sandy' Pentland. Deepmood: Forecasting depressed mood based on self-reported histories via recurrent neural networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 715–724. International World Wide Web Conferences Steering Committee, 2017.

[134] Pierre Thodoroff, Joelle Pineau, and Andrew Lim. Learning robust features using deep learning for automatic seizure detection. In *Machine learning for healthcare conference*, pages 178–190, 2016.

[135] S Tibken. Samsung, smartthings and the open door to the smart home. *cnet CES*, 2015.

[136] M. Torchia and M. Shirer. Tidc forecasts worldwide spending on the internet of things to reach $745 billion in 2019. `https://www.idc.com/getdoc.jsp?containerId=prUS44596319`, 01 2019. [Online; Accessed 5 Sep. 2019].

[137] Turker Tuncer, Sengul Dogan, and U. Rajendra Acharya. Automated accurate speech emotion recognition system using twine shuffle pattern and iterative neighborhood component analysis techniques. *Knowledge-Based Systems*, 211:106547, 2021.

[138] Georgios Varsamopoulos, Ayan Banerjee, and Sandeep K. S. Gupta. Energy efficiency of thermal-aware job scheduling algorithms under various cooling models. In Sanjay Ranka, Srinivas Aluru, Rajkumar Buyya, Yeh-Ching Chung, Sumeet Dua, Ananth Grama, Sandeep K. S. Gupta, Rajeev Kumar, and Vir V. Phoha, editors, *Contemporary Computing*, pages 568–580, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[139] Isabel Wagner and David Eckhoff. Technical privacy metrics: A systematic survey. *ACM Comput. Surv.*, 51(3), jun 2018.

[140] Xin-Cheng Wen, Jia-Xin Ye, Yan Luo, Yong Xu, Xuan-Ze Wang, Chang-Li Wu, and Kun-Hong Liu. Ctl-mtnet: A novel capsnet and transfer learning-based mixed task net for the single-corpus and cross-corpus speech emotion recognition, 2022.

[141] Thomas Winkler and Bernhard Rinner. Security and Privacy Protection in Visual Sensor Networks: A Survey. *ACM Comput. Surv.*, 47(1):2:1–2:42, May 2014.

[142] Shuochao Yao, Yiran Zhao, Huajie Shao, ShengZhong Liu, Dongxin Liu, Lu Su, and Tarek Abdelzaher. Fastdeepiot: Towards understanding and optimizing neural network execution time on mobile and embedded devices. In *Proc. of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018.

[143] Jiaxin Ye, Xin-Cheng Wen, Yujie Wei, Yong Xu, Kunhong Liu, and Hongming Shan. Temporal Modeling Matters: A Novel Temporal Emotional Modeling Approach for Speech Emotion Recognition. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, June 2023.

[144] Samuel Yeom, Matt Fredrikson, and Somesh Jha. The unintended consequences of overfitting: Training data inference attacks. *CoRR*, abs/1709.01604, 2017.

[145] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[146] Kan Yuan, Di Tang, Xiaojing Liao, XiaoFeng Wang, Xuan Feng, Yi Chen, Menghan Sun, Haoran Lu, and Kehuan Zhang. Stealthy porn: Understanding real-world adversarial images for illicit online promotion. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 952–966, 2019.

[147] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In *Thirty-second AAAI conference on artificial intelligence*, pages 2604–2611. AIII Press, April 2018. Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18 ; Conference date: 02-02-2018 Through 07-02-2018.

[148] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017.

[149] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23, 2016.

[150] Jianlong Zhou, Amir H. Gandomi, Fang Chen, and Andreas Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5), 2021.