Learning Representations for Anonymizing Sensor Data in IoT Applications

by

Omid Hajihassani

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering

University of Alberta

© Omid Hajihassani, 2021

Abstract

Recent years have witnessed a growth in the deployment of IoT devices in homes and workplaces. The number of IoT devices is projected to surpass tens of billions in the near future. This rapid growth can be credited to useful insights and convenience offered by IoT services and applications. A typical IoT device is equipped with one or several sensors which are capable of collecting high-fidelity and high-sample-rate data from the environment, often without notifying the user. This ubiquitous and inconspicuous data collection threatens user privacy as the collected data may contain private or sensitive information which can be extracted by malicious applications through unsolicited inferences. This thesis investigates potential solutions based on generative machine learning models to limit the accuracy of privacy-intrusive inferences with an imperceptible impact on the accuracy of useful and desired inferences.

We begin this thesis by surveying different approaches to privacy-preserving data collection and processing. As the first contribution of this thesis, we investigate the ability of variational autoencoder (VAE) models to learn representations that enable hiding the private information embedded in sensor data. Specifically, we modify the loss function of standard and conditional VAE models to obtain two different anonymization techniques. These techniques perform deterministic and probabilistic manipulations in the learned latent space of autoencoders. These manipulations effectively support data anonymization when the corresponding latent variable is used to reconstruct the original data.

To evaluate our methods, we use two publicly available Human Activity Recognition (HAR) datasets, namely the MobiAct and the MotionSense datasets. These datasets contain both public and private information about users which can be detected using inference models (desired and sensitive inferences, respectively). Our goal is to use the proposed techniques to conceal private information while maintaining the accuracy of the desired inference as much as possible.

We evaluate the efficacy of each technique in concealing private information through ablation studies and comparison with multiple baseline methods, including recent techniques proposed in the literature. We evaluate our techniques by treating the activity attribute in both datasets as public information, and the gender and weight of subjects as private information. We show that state-of-the-art anonymization techniques are vulnerable to a user re-identification attack, while our techniques are less susceptible to this attack thanks to the proposed non-deterministic manipulations. In comparison to the best autoencoder-based baseline method, we achieve 13.48% lower privacy loss on average in the two HAR datasets while getting a comparable activity inference accuracy. This indicates that a better trade-off between utility and privacy is achieved by our techniques. Moreover, we discuss how users can navigate the utility-privacy trade-off (according to their own needs and values) by tweaking the weights in the modified loss functions of the generative models. We show that one of the proposed anonymization techniques can simultaneously conceal multiple private attributes with only a small decrease in the anonymization performance.

Preface

This thesis is original work by Omid Hajihassani. Two chapters are based on publications co-authored by the author of this thesis. In particular, Chapter 3 is inspired by the paper published in the SenSys-ML workshop [22]. This paper presents the re-identification attack and proposes an anonymization technique based on a VAE. More recent results from [23] are added to Chapter 3. The author of this thesis was responsible for designing the algorithms, training the VAE models, analyzing the results, and producing figures.

Chapter 4 discusses the original work accepted in 2021 IoTDI conference. This work discusses supervised latent representation learning and manipulation using conditional VAEs. This article discusses different baseline methods for better comparison with the CVAE technique. The author of this thesis was responsible for the design and evaluation of the algorithms, implementing baseline methods and writing the paper.

Acknowledgements

Foremost, I am sincerely grateful to my supervisors, Professor Omid Ardakanian and Professor Hamzeh Khazaei, for their patient guidance and supervision. I am indebted to my supervisors for their dedication and careful evaluation of this thesis and all the published articles used as the foundation for this thesis.

I am genuinely thankful to my loving parents, who helped and supported my passion for knowledge. I am thankful to all good friends whose company kept me in good spirits.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 IoT and Pervasive Sensing Technologies

In recent years, there is an uptake in the use of Internet of Things (IoT)-enabled devices and services in homes and work environments. Their increasing popularity can be attributed to the valuable insights they offer. The IoT devices integrate various sensors and actuators enabling them to collect information about their surrounding environment and act on this information subsequently. We use the term "IoT-enabled technologies" to refer to systems and services that, in one way or another, use the backbone of the Internet to deliver insights, services, and control to home or enterprise users.

Nowadays, IoT devices span a wide range of consumer electronics, including virtual assistants (e.g., Alexa, Siri, or Google Home), smart thermostats, home security devices, and health monitoring devices. These devices can differ in shape, size, and the interface through which consumers interact with them. Nevertheless, they come with one or several sensors for monitoring the environment. The sensor data can be processed locally (on-device), be sent to third-party servers, or processed in the edge and cloud in a cooperative manner. The result of processing this time-series data is then returned to the user or stored for further analysis. This large-scale collection and processing of sensor data, which in many cases contains patterns that reveal consumers' private information, poses a privacy risk to IoT users.

With inconspicuous data collection, private information can be collected together with public and useful information. This poses a grave privacy threat to users, which can cause users to become wary of IoT-enabled technologies, and in turn, hinder the development and widespread use of numerous IoT devices. Consider a household that intends to use a smart thermostat to control their HVAC system. Despite the potential cost and energy saving benefits [27] the smart thermostat offers, users could abstain from installing it as they worry it may be capable of detecting the occupancy state of their households. The subsequent leak of occupancy information to malign parties can pose a significant privacy risk and even lead to thefts.

There are multiple other examples of the privacy risks associated with IoT devices. In a different context, IoT devices can be used to violate privacy of employees. An example of this is when a company utilizes sensor data and analytics to detect slackers [31]. This can indeed be beneficial to the company and its management, but can be regarded privacy-intrusive by the employees if done without their prior knowledge and consent.

It has been further shown that the collected data from accelerometer and gyroscope sensors in smartphones and smartwatches, apart from useful insights like activity tracking, can be used to infer the consumers' gender, age, weight, height, and race, which are considered private information [42], [43]. These examples show that their proliferation could endanger users' privacy despite numerous benefits of IoT services.

Researchers show that a large fraction of IoT users are unaware of private inferences that could be performed on their data collected by non-audio/visual devices which are part of their smart home solutions [71]. Consumers, by and large, prefer the provided benefits over the associated privacy risks with their IoT devices. In a similar study [44], it is stated that 10% of smart assistant users' recordings were unintentional and not user-invoked, and around 5% of these were from guests. These examples emphasize the need for consumer awareness about the privacy risks associated with these devices and further development of privacy-preserving data analytic techniques that inhibit intrusive inferences [66].

1.2 Trade-off between Data Utility and Privacy

We model the relationship between private information leakage and usefulness of the collected data by the utility-privacy trade-off. All anonymization solutions address this trade-off between the data utility and privacy loss. In the collected time series sensor data, the private and public information cannot be ideally disentangled and attributed to specific, identifiable data sections. Hence, any attempt to hide, remove, or sanitize private information from collected data eventually results in degraded data utility.

In particular, solutions that aim to sanitize the original data through random perturbations, masking, and down-sampling typically lead to significant reduction of data utility. Both the amount of this degradation and the sanitization process must be taken into account while studying different anonymization solutions. The most effective anonymization solution is the one that removes the greatest amount of private information from the data while keeping the utility of the anonymized data as high as possible.

1.3 Existing Solutions and their Shortcomings

Various techniques and solutions have been proposed in different contexts that aim to sanitize private information from the collected sensor data while respecting the aforementioned utility-privacy trade-off. These solutions range from the addition of noise and perturbations to data [14], [58] to using generative models to learn and manipulate private attributes of the data [42]. The solutions that are most suitable for the anonymization of big, public datasets include k-anonymization [58] and differential privacy [13]. They provide plausible deniability to end-users and are best suited for anonymizing users' identities in public datasets. Through k-anonymization, it is possible to guarantee that every entry in the dataset resembles at least k-1 other entries in that dataset. Hence, the possibility of pinpointing any single end-user given each data entry is greatly reduced. k-anonymity shows to be useful in de-ifentifying the dataset, even in cases where the attacker incorporates prior knowledge.

These techniques, including k-anonymity, are not suitable as on-device solutions deployed at the edge. Moreover, it is our assumption that the cloud servers that collect and then anonymize the datasets cannot be trusted, and it would compromise users' privacy if we send their data over the Internet to cloud servers. Due to these reasons, more secure on-device techniques are required to perform anonymization at the edge, which provide better anonymization performance compared to techniques such as k-anonymity.

On-device anonymization solutions can be widely categorized into operating system and hardware level solutions and generative models. The former is a category of solutions that include specialized techniques tailored to the hardware and software of edge devices to provide privacy to users and avoid data leakages. These solutions answer different privacy threats such as safeguarding against the membership inference attacks (MIA) [53] on deep learning models and user data leakages. If successfully performed, the MIA attack is shown to detect whether a given data has been part of the training set of a white-box or black-box neural network model. Authors from [46] propose using the trusted execution environments (TEE) in the ARM processor used in edge devices. This technique provides a resilient approach to executing (training and inference) of a neural network model. By showing that MIA accuracy is attributed to the few final layers of a neural network, the DarkNeTZ framework executes the last layers of each model in the TEE section of the processor to protect against the MIA technique.

There are multiple hardware and operating system level solutions that avoid the leakage of users' private information through unauthorized memory accesses or third-party applications with access to data. If third-party applications running on the edge device transmit users' data to cloud servers without the users' knowledge, it will endanger users' privacy by sharing their private information. Authors from [16] propose taint tracking combined with TEE to control applications' user data accesses. In this technique, by treating the sensors on edge devices as the source and the network interface card as the sink, taint-tracked data cannot move from any source to a sink without first going through a sanitization module. In the sanitization module, the FlowFence framework ensures to sanitize the sensitive parts of information before sending the data to the sink. Moreover, through sandboxed execution and memory access abstraction of sensitive data, FlowFence helps prevent direct access to users' sensitive data by third-party applications.

A common shortcoming of these hardware and operating system level solutions, is that even when techniques such as FlowFence are used, there is no guarantee that on-device sensitive inferences are not prevented. So, on-device data anonymization techniques are required to stop sensitive inferences. In this thesis, we propose using generative models to manipulate private attributes of the original data by first learning a latent representation of the data and then manipulating it. Data generative frameworks such as Generative Adversarial Network (GAN) [20], Autoencoder (AE), Variational Autoencoder (VAE) [29], or Conditional Variational Autoencoder (CVAE) [30] techniques have been recently successfully applied to data synthesis in different fields. GANs have been applied to image inpainting with contextual attention [68], medical image synthesis [48], or even creative tasks such as photo cartoonization [7]. Variational autoencoders have also been used for tasks where learning a representation of the input data is required. Compared to AEs, VAEs can help learn salient features that capture and mimic the true posterior distribution of the data.

Apart from using generative models in data synthesis such as synthesis of human motion activity [72] and medical images [48] that can help solve the lack of data in many research communities. Multiple researchers have explored the use of generative models to manipulate features before reconstructing the data. GAN and VAE frameworks can alter the features of their input images to make subtle changes in their reconstructed data. For example, authors from [34] show how it is possible to change and morph one input image subject's gender from male to female and vice versa.

In data anonymization, high data utility loss is not tolerable. Hence, techniques that are proposed need to preserve the intended data utility. Authors from [69] train an autoencoder that learns to inject noise and perturbations into the data to diminish the sensitive information inference accuracy on the anonymized data. This proves to be effective on two Human Activity Recognition (HAR) datasets in hiding the subjects ID, gender, height, and age. Like this work, instead of addition of noise and perturbations, authors from [42] use adversarial training for autoencoders to diminish the accuracy of private inference models. Authors evaluate their results on the MotionSense HAR dataset, where the Anonymization Autoencoder (AAE) technique essentially removes the gender identity of the users from the sensor readings.

In this thesis, instead of using perturbations or fixed adversarial models, we propose manipulations in the latent representations learned by a CVAE or a VAE to protect user data privacy. We discuss two manipulation techniques that change the private attributes of the reconstructed data such that the accuracy of privacy-intrusive inferences is degraded.

1.4 Research Questions and Contributions

This thesis builds on the previously proposed autoencoder-based generative models which despite being able to sanitize the reconstructed data, cannot withstand an attack referred to as the re-identification attack [22]. We hypothesize that this is due to the deterministic nature of the operations and the fixed adversarial models used in the previous anonymization techniques. By exploiting this deterministic nature, the adversary can learn to re-identify private attributes of the anonymized data.

We propose two different frameworks, one based on VAE in Chapter 3, and another based on CVAE in Chapter 4. We evaluate these two anonymization frameworks on two publicly available HAR datasets including the MotionSense [42] and MobiAct [60] datasets. We study multiple baselines that use mean manipulation with VAE and conditional variable modification with CVEA. The contributions of this work are as follows:

- We evaluate the ability of conditional variable modification with CVEA and mean manipulation with VAE in obscuring private attributes in the anonymized data. We further compare them with multiple baselines by performing experiments on two HAR datasets, namely MotionSense [42] and MobiAct [60].

- We present ObscureNet, our technique based on the CVAE framework, and show that the anonymization performed by ObscureNet outperforms all the other baselines and explain how the CVAE loss function can be modified to anonymize multiple private attributes instead of a single private attribute.
- We show that by tuning weight factors in the loss function of our anonymization methods, it is possible to achieve different trade-offs between the utility and privacy.

Below, we list four different research questions which we aim to answer in this thesis.

> *RQ 1. How successful are the two anonymization techniques considering data utility and privacy?*
>
> *RQ 2. What results do these techniques yield in terms of anonymization performance and vulnerability to the re-identification attack when using non-deterministic manipulations?*
>
> *RQ 3. Can VAE and CVAE-based techniques be used to trade privacy for data utility?*
>
> *RQ 4. Can our manipulation techniques be used to anonymize multiple private attributes all at once?*

We discuss possible solutions to these research questions in Chapter 6.

1.5 Roadmap

This thesis is organized as follows. In Chapter 2, we discuss the related work and the background knowledge of our work. Chapter 3 proposes the mean manipulation VAE-based anonymization framework. It introduces the modification of the VAE's loss function to include a classification loss that helps learn manipulation-friendly latent representations. Chapter 4 discusses our CVAE-based anonymization framework dubbed as ObscureNet. Chapter 5 provides our evaluation results and the details of the datasets used in our evaluation. We further outline different baselines and present our ablation studies. Chapter 6 concludes this work and discusses answers to our proposed research questions.

Chapter 2

Background and Related Work

In this chapter, we survey different techniques that are used to preserve user privacy in IoT applications. These techniques range from primitive methods such as direct, physical disruption of the data collection process to machine learning models used for data anonymization. The following gives an overview and a taxonomy of the related literature on anonymization and privacy-preserving techniques.

As this thesis aims to address on-device IoT privacy, in next section we mostly delve into the methods proposed for on-device data anonymization, and we forgo a detailed discussion about cloud-based privacy solutions. Here, we assume users do not trust cloud servers and third-party service providers.

2.1 Disrupting Physical Data Collection

Perhaps the most primitive and yet effective solution for addressing the IoT data privacy issue is the direct and physical disruption of the data collection process in sensor systems and IoT devices. One such privacy-preserving measure is the incorporation of a physical mute button on Google Nest Mini devices, which helps with promoting user privacy by enabling users to pause or disable data collection [18]. This proves to be 100% effective in cutting off the sensors' access to the environment and user data.

In [57], authors propose MicShield, a technique and a device for jamming microphones of smart voice assistants (VA). MicShield prevents the private speech of users from reaching VAs. MicShield generates jamming signals to shift the speech in audible range to inaudible speech. Another appealing example of physical disruption of the data collection process is proposed by [74]. To disrupt visual data collection from restricted areas and sights, a smart LED is used in [74] to generate flickering patterns. This way, the photograph of the scene becomes noisy and not usable. However, users must have control over the sensor or have the ability to meddle with the environment to accomplish this, hence, interfering with the data collection process is not always possible. Furthermore, most related work assumes that sensor data is already collected.

Despite being effective, such measures fail to become widespread, adaptable solutions for two reasons. One reason is that when the user decides to start using the device again to its fullest capacity, private data can be collected and processed. Moreover, adopting such solutions at a large scale is sometimes infeasible due to the lack of access to the IoT sensors.

2.2 Low-level Anonymization Techniques

This category of anonymization techniques includes fine-grained access-control mechanisms and isolated execution environments [35], [46], [49], [54], privacy-preserving and secure communication protocols [4], [15], [56], blockchain-based privacy schemes [6], [11], [52], federated learning techniques [33], [40], and hardware design choices [59], [64].

2.2.1 Access-control Mechanisms

These solutions aim to control and manage data flow between applications running on an edge device or over the Internet. The idea is to provide fine-grained and user-friendly access control mechanisms so that users have control over their collected data and determine what happens to it. These solutions combine different techniques, including operating system-level data abstraction, sandbox execution of third-party code, and taint tracking, to minimize user privacy loss.

Authors in [16] promote user privacy through sandboxed execution of developers' code in quarantined execution modules, and taint-tracked data handles.

To be specific taint-tracked opaque data handles help prevent applications from accessing sensitive data and sharing it via the network interface. This framework is referred to as FlowFence. This way, by applying taint to different parts of the collected data, a sink in this scheme (third-party application or network interface card) cannot read the raw, tainted sensor data without it being, first, sanitized through a sanitization module. The sanitization modules call on users' access preferences to check which part of the data is authorized to be sent to a sink. After the omission of specific sensitive information in the data, the data is sent to the sink. In [54], authors propose a privacy abstraction technique to manage access tussles to sensor data at the operating system level. Moreover, the authors suggest a mechanism for managing sensor data access tussles by leveraging a specific functionality and privacy trade-off.

The FlowFence framework in [16] accommodates indirect access to private and sensitive information in the data through the use of quarantined execution environments. Developers address the data stored in the secure memory of IoT devices by using opaque data handles that do not enclose data attributes to developers. In the execution phase, the opaque handles are resolved by the FlowFence framework in a secure execution environment, and real data values are kept from being exposed to third-party applications. Another technique that uses taint tracking is TaintEraser [73]. The TaintEraser tool provides an efficient and applicable solution to prevent information leakage through data tainting. The authors implement dynamic taint analysis techniques that enable the study of information flow from applications to the network and the local file system. By providing users with simple privacy policies, they can specify sensitive parts of the input data for the TaintEraser tool to replace them with random noise when data from these sensitive inputs is being written to the local file system or is being sent over the network.

Apart from FlowFence that proposes taint tracking for preventing sensitive information leakage in IoT devices, authors in [5] propose SAINT, a tool that provides static taint analysis for IoT use cases. In contrast to FlowFence, SAINT is proposed as the first approach that detects the flow of sensitive information between the IoT applications. SAINT proves to be able to identify sources and sinks in IoT applications by translating IoT applications source code into an Intermediate Representation (IR) and performing static analysis on the retrieved IR of each application.

Although all these solutions are robust and useful to prevent information leakage from sensitive information sources to third-party destinations, they fail to prevent unwanted and private inferences on the user data which are performed by third-party applications. Third-party applications can still run privacy-intrusive inferences on users' data and infringe the user privacy. Our goal is to prevent this by proposing generative model-based solutions to obscure private information in the collected sensor data before third-party applications process it. Our approach can be combined with taint tracking solutions to provide a comprehensive solution for preserving privacy.

2.2.2 Trusted Execution Environments

Multiple privacy-preserving techniques have adopted Trusted Execution Environments (TEE). As it is the case with the FlowFence framework [16], sandboxed execution and TEE are essential tools for providing privacy-preserving data analytics to users. TEE, in contrast to Rich Execution Environments (REE), provides an abstraction of operating system and hardware resources that are kept separate from the rest of the applications and resources running on a device. By providing secure registers, memory segments, and OS data structures, TEEs enable safe and secure execution of sensitive code. Intel's Software Guard eXtensions (SGX) [45] supports private and secure enclaved software execution of code, for example, cloud-based execution of secure DNN models [21], [24].

Given that most IoT devices use ARM-based CPU platforms, for TEE. We consider ARM TrustZone, which is an ARM solution to providing secure execution environments for mobile and edge devices. Each System-on-Chip (SoC) that comes enabled with TrustZone provides hardware resources for both normal and secure execution of programs. Similar to the concept of TEE, applications running in the trusted and secure zone are provided with secure resources. Recently, numerous IoT applications have benefited from ARM

TrustZone including StreamBox-TZ [50], TruZ-Droid [67], and SeCloak [35]. StreamBox-TZ [50] (SBT) is a stream analytics engine designed for IoT devices which provides user privacy. Moreover, TruZ-Droid [67] integrates TrustZone with the Android operating system to protect users' information from being sent to third-party servers and ultimately misused.

SeCloak provides a solution for on-off control of smartphone peripherals using ARM TrustZone and TEE [35]. By using TEE, SeCloak provides security and privacy even when the system is compromised. This framework uses a secure kernel alongside the operating system (Android or Linux), diminishing the need for code changes in the operating system. SeCloak provides reliable control of peripherals without inflicting costly performance overheads on the device.

The authors in DarkneTZ [46] propose a framework that utilizes the TEE resources provided by ARM TrustZone to protect the IoT models data privacy. Concretely, DarkneTZ is designed and proposed to ward of Membership Inference Attacks (MIA) on the trained inference models and DNNs downloaded from cloud servers for on-edge inferences. In MIA, which has white-box and black-box variants, the attacker can, to a reasonable extent, infer whether a given data point was part of the original training set or not [53]. Given the limited resources of TEEs and the increasing size of modern DNNs, not all layers of the model can be executed in the TEE of the edge device. Hence, the main contribution of DarkneTZ is studying how, by executing only the final layers of the DNN models in the TEE, the MIA's precision can be diminished.

It is shown that the final layers of a neural network are more sensitive to the MIA. Hence, the precision of the MIA can be significantly reduced by executing the few final layers of a DNN model in the trusted zone. This way less execution overhead is incurred on the system. All in all, Trusted Execution Environments enhance the privacy and security of data in IoT devices to a great extent. However, TEE cannot stop local or cloud-based sensitive and privacy-intrusive inferences into the users' data, as was the case with access control mechanisms. This is because malicious applications can still perform privacy-intrusive inferences on users' data. This calls for the design of better techniques that protect user data privacy against sensitive inferences.

2.2.3 Privacy-preserving Protocols

In [4], an efficient privacy-preserving querying protocol is proposed for sensor networks, assuming that client queries are processed by servers controlled by multiple mutually distrusting parties. Client queries reveal both an array of specific sensors of interest to each client and the temporal relationships between subsequent queries. The privacy risks will dissuade organizations from sharing resources to build large-scale shared sensor networks. To address these risks, the authors propose the SPYC protocol [4], which guarantees privacy protection if query processing servers do not cooperate to attack the clients' privacy. They also discuss possible solutions when servers cooperate to infringe the privacy of clients.

The SPYC protocol has been proposed for large-scale sensor networks and processing of these queries. In [56], a new cost-effective, secure, and privacy-preserving protocol for Smart Home Systems (SHSs) is proposed. By eavesdropping on the network's communication an attacker can infer who the SHS user is and when the devices are used. This is done by analyzing the frequency and type of communication from SHS IoT devices to their central controller. This fact can lead to grave security and privacy implications that might cause break-ins or life threatening situations. The authors in [56] outline two of the main challenges in designing SHS communication schemes namely privacy and efficiency concerns. To ensure users' privacy and security, the authors propose chaos-based cryptography and one-time key generation using well-known chaotic systems and Message Authentication Codes (MACs) for encrypted and secure data transmission.

The backbone of generic SHSs is comprised of Radio Frequency Identification (RFID) tags, sensors, central controllers, and monitoring interfaces [56]. The authors in [15] propose a secure and private RFID communication protocol and study the security implications of using RFID tags in the medical context. Unlike many others, the proposed protocol provides tag anonymity, which is a step toward patient privacy and is shown to be in line with medical and hospital security standards. This protocol is shown to be resistant to typical RFID attacks, including the Replay, Synchronization, and DoS attacks.

Although communication security of IoT devices in SHSs and sensor networks is a critical and indispensable part of any IoT security scheme, secure communication protocols cannot solely address privacy concerns given data pervasive and inconspicuous data collection by IoT devices in our living environments.

2.3 Blockchain-based IoT privacy

Several research teams are currently investigating blockchain (BC) technology and its applications to IoT security and privacy [6], [10], [11], [52]. Blockchain offers three major benefits [52]. One advantage of using BC as the framework for designing SHS and other IoT networks is that multiple miners in the network verify the generated sensor data's authenticity before adding the data record into the ledger. This decentralized verification and authentication of sensor readings before adding them to the dataset helps the IoT system to ward off several security attacks, including data manipulation attacks from malicious actors [52].

Another security and privacy contribution of BC to IoT is that the accepted and added data to the ledger cannot be tampered with and changed. The third and another essential attribute of BC augmented IoT networks is the absence of central authority and storage servers [52]. In this scheme, malicious and adversary nodes can be detected and identified by miners and then terminated.

The authors in [12] propose a BC-based smart home architecture by proposing modifications to the Bitcoin cryptocurrency for security and privacy of IoT devices. Although the efficacy of the proposed BC-based technique is exemplified in the scenario of smart home IoT devices, the authors claim that their proposed methodology is application-agnostic and can adapt to new IoT schemes and use cases. The main contribution of this technique is in the conversion of the conventional Bitcoin BC to a lightweight instantiation, which is suitable for resource-constrained IoT devices. The authors eliminate the need for proof of work when submission is happening to the local Immutable Ledger (IL). This reduces the overhead and computation demand on the edge devices and sensors. The authors introduce the smart home tier, which consists of all local devices or overlay nodes that generate transactions with the locally managed IL. The IL is kept in the smart home manager or SHM, which can be a resourceful mobile device or even the homeowner's personal computer. The SHM then uses the policy header in the local IL defined by the homeowner to authorize the received transactions.

The authors in [52] point out the fact that using symmetric key encryption and decryption schemes such as AES would lead to privacy and confidentiality issues concerning the data shared by the IoT networks as miners can have access to the collected data. The authors propose using attribute-based encryption (ABE) to encrypt the data shared in the BC-enabled IoT network. In the proposed attribute-based encryption scheme, the data owner is referred to as the cluster head, which uses ABE to anonymize the data received from sensors. The cluster head aggregates data and encrypts the data using the attribute-based encryption scheme by defining a set of attributes so that the miners can see and verify the transactions. As an example, the cluster head can define attributes such as "DOCTORS" or "NURSES" so that only miners having the "DOCTORS" or "NURSES" attribute can decrypt, verify, and use the appended data in the blockchain.

These proposed blockchain-based IoT architectures are suitable for secure and confidential storage, communication, and decentralized verification of sensor data generated by distributed sensors. However, just like the privacy-preserving techniques discussed earlier, the proposed blockchain-based IoT techniques fail to address privacy-intrusive inferences made by third-party applications and servers on the user data.

2.4 Hybrid and Federated Learning

The training process of inference models in the IoT environment and the execution of the trained models on edge devices with limited resources call on different computing paradigms to make sure user privacy is preserved with a reasonable computation overhead. Different techniques are proposed to address these issues by discussing different computing paradigms such as cooperative edge and cloud computation [49] and federated learning techniques [33]. In the latter, the possibility of distributed or decentralized training of inference models is provided while diminishing the need for publicly available data. Works such as [40], [70] combine the idea of federated learning with BC technologies to provide convenient, secure, attack-resilient, and privacy-preserving decentralized model training processes.

Specifically, in [70], the authors propose a technique using federated learning and BC technology to eliminate the need for central data and model aggregators by replacing them with trusted miners in BC. In this scheme, the consumer uses resourceful local edge devices such as mobile phones or mobile edge computing (MEC) servers to train an initial model. This initial model is signed and sent to the blockchain. This way, it can easily protect the process against malicious consumers as all the transactions are traceable to the malicious party. To provide data privacy and further protection for users, the authors in [70] enforce differential privacy through the addition of noise to the consumers' data before the local training process for the initial model begins. The improvised data perturbation is performed through a normalization step based on [28].

In the domain of Industrial Internet of Things (IIoT), the concept of private multi-party data sharing has drawn a lot of attention [33]. The goal of multi-party data sharing is to address storage and computation constraints by sharing data from multiple distributed data owners. In [33], the authors note that most of the published work in this domain endanger consumers' privacy by using central data curators. Instead, in [33], permissioned blockchain, which is a differentially private multi-party data model sharing method, is proposed that shares the models learned by federated learning instead of raw data. Essentially, the proposed method turns the data-sharing problem into a machine learning one. This is done by using federated learning techniques integrated with differential privacy to protect privacy of the collected data. Moreover, the need for a central data curator is eliminated by using blockchain technology's collaborative architecture.

In addition to distributed model training and data sharing practices, distributed computing paradigms can be used for inference and model execution on both the edge devices and servers. With the growing size and computational demand of inference models that are running on edge devices, the authors in [49] propose a methodology on how to safely and securely divide the computation process between edge devices and servers while ensuring user data privacy. By fine-tuning the trained models with their suggested Siamese architecture, this technique preserves users' data privacy while achieving a reasonable inference accuracy. Authors in [65], propose using an encoder-decoder architecture in compressing the data sent from the edge to the cloud. DeepCOD uses an encoder-decoder architecture with a simple encoder and a deep decoder neural network where the goal is to achieve the best inference results rather than precisely reconstructing the input data.

2.5 Cryptographic Techniques for Enhancing Privacy in Data Analytics

Cryptographic data security and privacy approaches are perhaps the most common solutions for secure and private data storage and transfer. The application of cryptography to IoT security and privacy has recently gained traction. This includes lightweight hardware cryptosystems [51], [59], [64] and Homomorphic encryption [63] for Neural Networks. In [59], authors propose SecureData, a secure encryption scheme for IoT secure data collection. SecureData provides secure data collection measures using a lightweight FPGA implementation of KATAN [8]. KATAN [8] is a hardware-optimized block cipher with an 80-bit key size. SecureData uses the KATAN secret cipher in the network data transmission layer. The authors in [59] evaluate and show the efficacy of SecureData in providing data security and privacy for healthcare IoT use cases with various threat models, including eavesdropping and data leakage scenarios.

The authors in [64] present a survey of hardware-based optimization of cryptosystems in the IoT security and privacy area. These works include lightweight implementations of the widely used AES block cipher, techniques to prevent side-channel attacks, and pseudo and true random number generators (PRNGs and TRNGs) for ultra-low-power (ULP) devices. Apart from secure IoT data collection and communication, privacy of core IoT technologies have been extensively studied. For example, the widely adapted RFID technology requires a robust authentication protocol to help eliminate privacy and security risks and attacks, such as the Replay attack. The authors in [51] propose the Gossamer protocol for secure and private authentication of lightweight RFID devices.

Apart from hardware-based optimization and secure and private data collection protocols, techniques such as Homomorphic encryption [19] are being studied for their use in secure DNN execution on edge and in cloud platforms. Homomorphic encryption allows performing operations on encrypted data without the need for decrypting it first. This means that there is no longer a need for decryption of the ciphertext from consumers to perform data analytics. The ingenuity of this encryption scheme is that the decrypted output of operations on the encrypted data is the same as the output of operations on the original, unencrypted data. Reference [63], introduces the idea of Crypto-Nets, which discusses deep neural networks that, with the help of Homomorphic encryption, allow for operations on encrypted user data rather than the original user data. If a hybrid execution paradigm is selected for the cooperative execution of models on edge and cloud servers, the need for decrypting the user data is eliminated. However, due to its high resource demand, Homomorphic encryption has not seen widespread use on ultra-low-power edge devices.

Despite the benefits of these security measures and the necessity of adopting some of these measures (at any cost) for data security, their need for making drastic design changes and their high computational costs render these techniques unpopular. We argue that although it is vital to have secure and encrypted communication and data storage on edge and cloud servers, these measures alone do not guarantee user privacy. The fact that third-party applications and services can still misuse user data underlines the need for adaptive privacy-preserving solutions.

2.6 Differential Privacy and k-anonymity

Differential privacy [13], [14] and k-anonymity [58] can be categorized as algorithmic solutions to user data privacy-preservation problem. Differential privacy allows service providers and third-party applications to collect information from users to build and further develop new systems without compromising the privacy of the users [14]. In differential privacy, a carefully tailored perturbation is added to the personal identifier information in the dataset, which provides privacy guarantees for owners of the data entries. This way, differential privacy enables the publication of aggregated datasets, thereby paving the way for further research and development in needed areas.

In most previous work, differential privacy has been used as a further guarantee of user data privacy (see for example [33], [70]). As we have already discussed attacks, such as the model inversion attack [17] or the membership inference attack [53], endanger the privacy of users in the dataset used for training the inference models in the deep learning community. Specifically, the attacker can identify whether a given data point has been used for training [53] or obtain the whole or parts of the training dataset [17]. To protect against such attacks, reference [1] proposes the use of differential privacy techniques to provide privacy guarantees to deep learning models.

Another technique that can be applied to users' privacy protection in aggregated datasets is the k-anonymity technique [58]. k-anonymity provides plausible deniability to data owners in a dataset by ensuring that each data entry's attributes and features in the dataset resemble at least k-1 other data entries. This way, data owners cannot be readily re-identified, while data still holds its utility. Other variants of k-anonymity include ℓ-diversity [41] and t-closeness [36]. The authors in [39] propose a new technique which uses clustering-based k-anonymity for sharing data from wearable devices.

These techniques show great promise in promoting privacy when sharing public data by mitigating threats such as user re-identification. However, as we will see in the next chapters, when private attributes are present in time series data, it is not feasible to pinpoint specific attributes in the recorded sensor data that directly contribute to sensitive information. Thus, these techniques cannot prevent sensitive inferences.

2.7 Generative Models

Deep generative models, including GANs [20], Autoencoders, and Variational Autoencoders [29], are being extensively used to generate a realistic version of an image which has a few differences with the original version (e.g., an image that has a different color) [32], [34]. Apart from image synthesis, generative models have been utilized to produce synthetic time series datasets [2], [37], [38], [61]. For instance, in [2] a Wasserstein GAN is used to generate balanced and realistic sensor data for HAR. In a recent study [38], the authors propose a framework based on GAN, called DoppelGANger, to generate network time series data with 43% higher fidelity compared to other baselines. Moreover, variants of autoencoders are commonly used to learn useful representations, especially when multiple sensing modalities are present. For example, autoencoders are used in [47] to learn a shared representation between multiple modalities.

Many studies utilize generative models to provide different levels of sensor data anonymization in IoT devices. Thus they can be regarded as on-device privacy-preserving solutions. These algorithmic solutions rely on machine learning techniques that use deep neural networks (DNN), and generative models such as generative adversarial networks (GAN) [20] and variational autoencoders (VAE) [29].

For instance, the use of deep generative adversarial networks for full-body and face de-identification in images is proposed in [3]. In particular, subjects' faces in an image are replaced by other faces generated by a deep generative network so that the camera feed can be used in applications such as activity detection, while protecting the identity of subjects in the image. But if the camera feed was meant to be used in biometrics face recognition, this data anonymization technique would not be suitable because of the distortion of facial features of the subjects. In a recent line of work [62], the application of GANs to face de-identification is discussed. Several studies also focus on using neural networks to anonymize patients' data in the public health domain. An automated system based on recurrent neural networks (RNNs) is proposed in [9] to de-identify patient notes (by removing protected health information).

Inspired by advances in deep generative models, recent works on data anonymization [42], [43] use autoencoders to reconstruct the input data such that private attributes are no longer identifiable. This approach provides a reasonable trade-off between data utility and privacy by minimizing the leak of private information while preserving the utility information content of the input data. However, the data anonymized by these networks is shown to be susceptible to the re-identification attack [22].

In [26], segments of time series data that can be used for sensitive inferences are black-listed, while other segments that can be used to make desired inferences are white-listed. The authors propose the Generative Adversarial Privacy (GAP) framework to offer a trade-off between utility and privacy. Replacement Autoencoder [43] builds on this idea by adding grey-listed inferences, i.e., non-sensitive inferences, to the white-listed and black-listed inferences introduced in [26]. These techniques are more suitable for replacing activities that are privacy intrusive, for example, smoking and drinking. According to our evaluations, race, gender, or other private attributes cannot be obscured using these techniques as white-listed segments also carry information about these attributes.

In [69], anonymization is performed through learning perturbations in transforming raw sensor data. The goal of these transformations is to reduce the sensitive data inference accuracy while maintaining the accuracy of desired inferences as high as possible. The authors refer to private attributes as *style* and public attributes as *content*. A transformation is used to map the style of a time series to random noise.

None of the above techniques enforce structure into the latent representation of autoencoders and utilize it to control data attributes in the synthesis process. In Chapters 3 and 4, we show that by using VAEs to learn and manipulate latent representations, we can achieve various anonymization techniques that differ in how they modify the private attributes.

2.8 Representation Learning with Variational Generative Models

We present two autoencoder architectures below, and give a mathematical derivation of the loss function in each case. The main distinction between these architectures lies in their ability to impose structure into the latent space. The reason we focus on variational autoencoders rather than GANs is that they represent the original data distribution more faithfully and provide a simple way to map data to its latent representation, which can be manipulated to anonymize data.

Let $\mathcal{X}$ be the domain of fixed-length embeddings of time series data generated by one or several sensors, and $\mathcal{Y}$ and $\bar{\mathcal{Y}}$ be respectively, domains of private and public attributes associated with embeddings in $\mathcal{X}$. Our dataset, $\mathcal{D} = \{(x_1, y_1, \bar{y}_1), ..., (x_m, y_m, \bar{y}_m)\}$ consists of m data embeddings, each denoted by x_i, and their corresponding private and public attributes denoted by y_i and $\bar{y}_i$. We assume this dataset is publicly available, and can be used by anyone to train models for desired and sensitive inferences[1]. We consider categorical attributes such as mood, activity, and gender. Hence, the private attribute takes value from $\mathcal{A} = \{a_1, \cdots, a_K\}$ and the public attribute takes value from $\mathcal{B} = \{b_1, \cdots, b_{\bar{K}}\}$.

2.8.1 Vanilla Variational Autoencoder

A VAE is an autoencoder comprised of a probabilistic encoder and a probabilistic decoder which are instantiated as two neural networks. The probabilistic encoder $q_\theta(z|x_i)$ maps sensor data x_i (or an embedding of it) to a distribution (e.g., a multivariate Gaussian) over low-dimensional continuous latent repre-

[1] Preventing the membership inference attack is outside the scope of this thesis.

sentations from which x_i could have been generated. The probabilistic decoder $p_\phi(x_i|z)$ produces a distribution over x_i given its latent representation z. This model can be used to generate a new version of the sensor data denoted by $\tilde{x}_i$. Note that θ and ϕ are network parameters that can be learned jointly.

Instead of maximizing the typically intractable marginal likelihood, VAE is trained to maximize a lower bound on the marginal log-likelihood which is known as the evidence lower bound (ELBO) [29]. This lower bound can be written for an individual data point denoted by x_i as follows:

$$\mathrm{ELBO}_i(\phi,\theta) = \mathbb{E}_{z\sim q_\theta(z|x_i)}\log p_\phi(x_i|z) - \mathrm{D}_{\mathrm{KL}}\big(q_\theta(z|x_i)||p(z)\big) \tag{2.1}$$

The Kullback–Leibler divergence term in ELBO acts as a regularizer for the approximate posterior. In the training phase, we maximize the sum of $ELBO_i$ over all samples in $\mathcal{D}$. This ensures that the encoder maximally preserves the information content of the input data and the decoder produces data as close as possible to its original input.

2.8.2 Conditional Variational Autoencoder

While variational autoencoders are suitable for learning unsupervised latent representations of data, the learned latent variables cannot be explained or mapped to salient attributes of input data. Learning useful latent variables which correlate to specific attributes in the dataset has received a lot of attention in recent years [30], [32]. Several efforts have been made to date to incorporate structure into latent representations in a supervised or semi-supervised fashion. Works such as [30], [55] introduce structure in the latent space by conditioning latent variables on data attributes. This can be accomplished by directly incorporating these features into latent representations in conditional VAE (CVAE). Specifically, a CVAE conditions the encoder, the decoder, or both on random variables representing data attributes. Thus, the probabilistic encoder and decoder can be written as $q_\theta(z|x_i,c)$ and $p_\phi(x_i|z,c)$, where the condition c can be a certain attribute of input data that we wish to encode. For example, it can be the private or public attribute(s) associated with an individual data point in a labeled dataset.

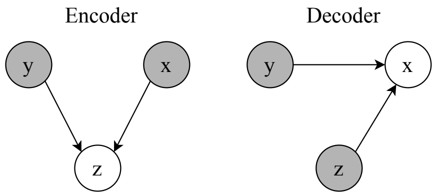

Figure 2.1: Graph diagram of encoder and decoder of a CVAE.

The variational lower bound of a CVAE can be derived from (2.1). The lower bound for an individual data point can be written as:

$$\mathbb{E}_{z\sim q_\theta(z|x_i,c)}\log p_\phi(x_i|z,c) - \mathrm{D}_{\mathrm{KL}}\big(q_\theta(z|x_i,c)||p(z)\big) \tag{2.2}$$

This objective is maximized using a stochastic optimization method to train the autoencoding model. In practice, the condition and learned latent variables can be concatenated before they are passed to the decoder for reconstructing the data. Figure 2.1 shows the encoder and decoder of a CVAE with condition variable, y.

In Chapters 3 and 4, we propose two different approaches that build on the VAE and CVAE frameworks that manipulate the latent representations learned by these models to change the private attributes of the original data in order to anonymize and obscure these private attributes.

Chapter 3

Unsupervised Learning and Manipulation of Latent Variables

3.1 Augmenting VAE with Classification Loss

This section describes how we learn a useful representation for an embedding of time series data generated by a sensor using a VAE with a modified loss function. We explain the details of the Mean Manipulation technique and discuss how manipulating the VAE's learned latent representations can transform the reconstructed data's private attributes. These steps are illustrated in Figure 3.1. Instead of using a single general VAE, we use multiple attribute-specific VAEs, one per each public attribute class. We further explain the idea of attribute-specific VAEs in Section 3.1.3.

3.1.1 Manipulating Latent Representations

Suppose a VAE is trained on $\mathcal{D} = \{(x_1, y_1, \bar{y}_1), ..., (x_m, y_m, \bar{y}_m)\}$ introduced in Section 2.8. The VAE maps each input embedding to its corresponding latent representation. Here, each latent representation is a vector of multiple latent variables. If we exactly knew which latent variable or a group of latent variables correspond to a given private attribute, we would be able to change these attributes before the decoder reconstructs the new version of data. This subsequently would change the private attribute of the reconstructed data and anonymize the data. Unfortunately, this is not possible because our VAE's la-

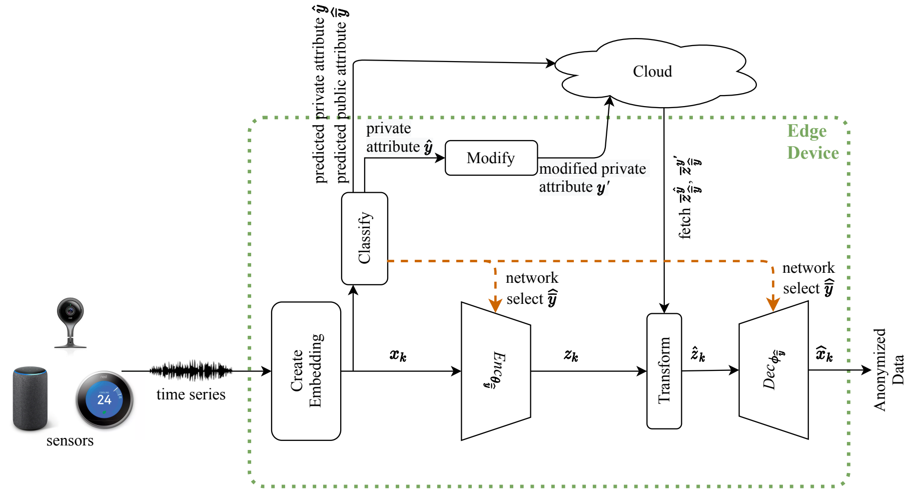
predicted private attribute $\hat{y}$
predicted public attribute $\widehat{\overline{y}}$
Cloud
Edge Device
private attribute $\hat{y}$
Modify
modified private attribute y'
fetch $\overline{z}^{\hat{y}}_{\widehat{\overline{y}}}$, $\overline{z}^{y'}_{\widehat{\overline{y}}}$
Classify
network select $\widehat{\overline{y}}$
network select $\widehat{\overline{y}}$
sensors
time series
Create Embedding
x_k
$Enc_{\theta_{\widehat{\overline{y}}}}$
z_k
Transform
$\hat{z}_k$
$Dec_{\phi_{\widehat{\overline{y}}}}$
$\widehat{x}_k$
Anonymized Data

Figure 3.1: Data anonymization on an edge/IoT device using the proposed anonymization technique. The mean latent representation for each pair of private and public attribute labels is assumed to be stored in a central (cloud) server.

tent representations are learned in an unsupervised fashion, and pinpointing one or a group of latent variables that correspond to each private attribute is not feasible. Hence, to perform anonymization, we have to modify all the latent representation variables by performing a translation in the latent variable space.

This idea is at the core of the mean manipulation technique proposed in our work [22]. To perform anonymization, we calculate the center of mass for latent representations of all samples in $\mathcal{D}$ which have the same pair of public and private attributes. These mean latent representations are then used to manipulate the latent representation of input data before it is passed to the decoder.

In the mean manipulation technique, by translating each latent representation from one region of the latent space to another and then reconstructing that through the VAE's decoder, we ideally change and manipulate the private attributes in the data public attributes are not altered.

The transformation of a latent representation in the mean manipulation technique involves a sequence of simple arithmetic operations. Consider a latent representation z_k with public attribute $\bar{y}$ and private attribute y, and

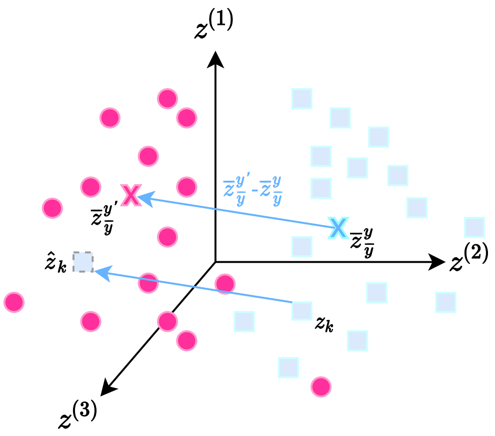

Figure 3.2: Overview of the mean manipulation anonymization technique assuming a 3-dimensional latent space. Latent representations which have a public attribute other than $\bar{y}$ are not shown in this figure.

let us denote the average of all latent representations with public attribute $\bar{y}$ and private attribute y by $\bar{z}_{\bar{y}}^{y}$. We obtain the transformed representation of x_k, denoted $\hat{z}_k$, by subtracting $\bar{z}_{\bar{y}}^{y}$ from z_k and adding $\bar{z}_{\bar{y}}^{y'}$ to the result. The probabilistic decoder takes $\hat{z}_k$ instead of z_k to reconstruct the data. We refer to $\bar{z}_{\bar{y}}^{y'} - \bar{z}_{\bar{y}}^{y}$ which is the Euclidean distance between the average of all representations with private attribute y and public attribute $\bar{y}$ and the average of all representations with private attribute y' and public attribute $\bar{y}$ as the *transfer vector*.

Figure 3.2 illustrates the transfer vector in a three-dimensional latent space. The markers show only the latent representations of embeddings with public attribute $\bar{y}$. Circles and squares represent data embeddings with private attribute classes y' and y, respectively. The mean latent representation is shown as a cross in each case. Once the transfer vector is found, it can be applied to modify a given data embedding's private attribute.

The mean manipulation technique can lower the accuracy of intrusive inferences but at the cost of reducing the accuracy of desired inferences. In the next two sections, we first discuss how the loss function of the VAE can be modified to learn anonymization-friendly latent representations. Later, we discuss how using multiple public attribute-specific VAEs helps with dealing with the class imbalance problem.

3.1.2 A Modified Loss Function for VAE

The loss function we use in this chapter builds on the original VAE's loss function proposed in [29] and in Equation 2.1. We modify the said loss function by adding an extra loss term that corresponds to the private attribute classification error, i.e., $Enc_\theta + f_\eta$. Essentially, the encoder network is supplemented with a classification layer, f_η, to encourage learning representations that are more representative of the private attributes associated with the input data. We first introduce this loss function and then discuss why minimizing this loss function can result in a more effective anonymization. The augmented loss function can be written as:

$$-\sum_{k=1}^{K}\Bigg(\mathbb{E}_{z_k\sim q_\theta(z_k|x_k)}\left[\log p_\phi(x_k|z_k)\right] - \beta\ D_{\mathrm{KL}}\big(q_\theta(z_k|x_k)||p(z_k)\big) \\ +\alpha\ \sum_{i=1}^{M} y_k^i \log\left(f_\eta^i(z_k)\right)\Bigg) \tag{3.1}$$

where z_k denotes the latent representation of the k^{th} input data embedding, y_k denotes the true private attribute class label of that embedding[1], and f_η is the classification layer.

The learned distribution over latent representations given x_k can be a multivariate Gaussian or a Bernoulli distribution. In our case, we choose a multivariate Gaussian since we are dealing with real-valued data. Note that the first two terms in this loss function are the two terms in Equation (2.1). The only difference is the introduction of the β weight factor for the Kullback–Leibler divergence term as explained in [25].

The main limitation of the β-VAE's loss function for data anonymization is the inherent trade-off between the quality of the reconstructed data and the disentanglement of the learned latent representations. In general, lower β values would yield better accuracy in the data reconstruction task (higher data utility), and higher β values would train the VAE to generate more disentangled latent representations (lower private attribute inference model accuracy).

[1] Note that y_k^i is 1 if and only if x_k belongs to the private attribute class a_i, and is 0 otherwise.

The best anonymization performance by a VAE is achieved when the data utility is the highest and the accuracy of the private inference model is the lowest. Thus, we need to tweak the loss function to have the highest data utility in the anonymized data (determined by the reconstruction loss and KL-divergence), while having as much disentanglement as possible (determined by KL-divergence) for the lowest private inference accuracy. As discussed in [25], there is a limit to the learning capacity of a conventional VAE's loss function. Hence, to increase the anonymization capability of the trained VAE, we add the private-attribute classification loss to the ELBO. Specifically, we use the latent representation of the original input data as input to a single-layer neural network, which infers each data embedding's private attribute class. This neural network, represented as f_η, will be trained alongside the VAEs encoder. In essence, the classification layer, f_η, and the VAE's encoder together form a classification network. As a result, the learned latent variables become more representative of private attributes in the data.

We use the cross-entropy loss, which is the distance between the predicted private attribute class of each anonymized data embedding and its ground-truth value, y_k^i [1]. We create a simple classification layer that maps the latent representations generated by VAE to the private attribute class labels of each of the corresponding input data entries as illustrated in Figure 3.3. Thus, the addition of the classification loss to the loss function encourages the VAE to learn more anonymization-friendly representations.

We argue that adding the classification layer, f_η, will force the probabilistic encoder to learn latent representations that are separable along the private attribute class labels, y_k's. Our results confirm that the added term to the objective function improves the performance of VAE in the anonymization task by introducing structure and enforcing a clear separation between different classes in the latent space. We instantiate f_η as a single layer of neurons with a softmax activation function. In particular, this layer contains K neurons, $n_0, ..., n_{K-1}$, where K represents the number of private attribute classes in the original dataset. The trainable set of weights used by the classification layer is denoted by η. Suppose each latent representation is a vector of J latent

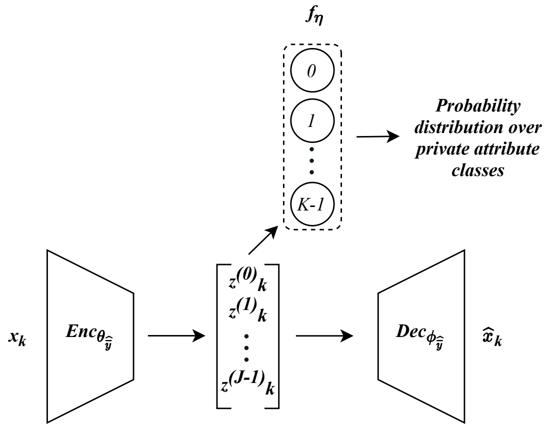

Figure 3.3: Variational Autoencoder with an additional classification layer denoted by f_η.

variables, $z_k^{(0)}, \cdots, z_k^{(J-1)}$. Thus, each η_j^m is the weight connecting input z_k^j to neuron n_m. The output of the m^{th} neuron in the classification layer can be written as $z_k \eta_m^\top$. The output of all the neurons goes through softmax activation to produce a probability distribution over private attribute classes given the input data: $f_\eta^m(z_k) = \frac{e^{z_k \eta_m^\top}}{\sum_i e^{z_k \eta_i^\top}}$.

The two hyperparameters in Equation (3.1), namely α and β, must be tuned for each VAE as discussed later in Chapter 5. The VAE and the classification layer are depicted in Figure 3.3.

3.1.3 Representation Learning with a VAE Customized for Each Public Attribute Class

By having attribute-specific VAEs instead of just one general VAE, which learns latent representations for all input data regardless of their public and private attribute classes [22], we break down the model into multiple models that are smaller in size. Each of these models is trained to reconstruct data for a given public attribute class.

One key advantage of using public attribute-specific VAEs is the reduction in the size of the model. It also allows for applying a higher disentanglement constraint (i.e., the weight β) in the training process. We get a 12-fold reduction in the model size and the number of trainable weights, from roughly 24 million weights in the case of a general VAE to a total of 2 million weights for all attribute-specific VAEs in the MotionSense dataset. Moreover, using parsimonious models enhances the anonymization performance when compared to [22].

Since we have multiple public attribute-specific VAE models, it is necessary to predict the public and private attribute classes of a given data embedding at the anonymization time. This information is used to determine which VAE must be selected for anonymization. We do this using the pre-trained classifiers shown in Figure 3.1.

3.1.4 Transforming Latent Representations

Algorithm 1 shows different steps of the proposed anonymization technique assuming that a VAE is already trained for each class of the public attribute. This algorithm operates on fixed-size embeddings of the input time series data. These embeddings are created by considering a window that contains a number of consecutive data points in the time series data. After the first embedding, a new embedding is created after a certain number of new data points are received (determined by the stride length).

Suppose, we have k data embeddings, denoted by $x_1, ..., x_k$. Each embedding has corresponding public and private attributes. Our proposed algorithm takes as input an embedding along with encoder and decoder parameters of different VAEs, and the average latent representation denoted by $\bar{z}^{y}_{\bar{y}}$ for each public attribute class $\bar{y}$ and private attribute class y. These average representations are calculated from the training dataset in the cloud or at the edge provided that the IoT device retains a copy of the training dataset.

In the next step, it loads the pre-trained public and private attribute classifiers (not to be confused with f_{η}) which are used to identify the predicted public attribute class $\hat{\bar{y}}$ and the private attribute class $\hat{y}$ for each data embedding x_k. After inferring the public and private attribute classes, we load $Enc_{\theta_{\hat{\bar{y}}}}$ and $Dec_{\phi_{\hat{\bar{y}}}}$ models for the predicted public attribute class $\hat{\bar{y}}$. The encoder part of this attribute-specific VAE encodes x_k in a probabilistic manner. The corresponding latent representation, z_k is achieved from the encoder. Once we have the representation, we change the inferred private attribute class label of

Algorithm 1: Anonymization with representation learning and transformation

```
Data: data embedding $x_k$, average latent representations,
      autoencoder parameters $\theta_{\bar{y}}$ and $\phi_{\bar{y}}$ for each public attribute
      class $\bar{y}$, pretrained classifiers for public and private attributes
Result: anonymized data embedding $\hat{x}_k$
$\hat{y}, \hat{\bar{y}} \leftarrow$ Classify($x_k$);
$z_k \leftarrow \mathrm{Enc}_{\theta_{\hat{\bar{y}}}}(x_k)$;
$y^{'} \leftarrow$ Modify($\hat{y}$);
$\bar{z}^{\hat{y}}_{\hat{\bar{y}}}, \bar{z}^{y'}_{\hat{\bar{y}}} \leftarrow$ Load mean latent representations;
$\hat{z}_k = z_k - \bar{z}^{\hat{y}}_{\hat{\bar{y}}} + \bar{z}^{y'}_{\hat{\bar{y}}}$;
$\hat{x}_k \leftarrow \mathrm{Dec}_{\phi_{\hat{\bar{y}}}}(\hat{z}_k)$ ;
```

x_k via a simple function which we refer to as `Modify`. This function converts the predicted private attribute class label of x_k from $\hat{y}$ to an arbitrary private attribute class label denoted by $y^{'}$.

We note that the `Modify` function can be either deterministic or probabilistic. When the private attribute is binary, the deterministic modification converts one class label to the other one at all times. When the private attribute class is not binary, an arbitrary bijective function can be used. In the case of probabilistic transformation, a probabilistic modification function is used. Specifically, for each data embedding, we decide whether to perform the mean manipulation based on a cryptographically secure stream of pseudo-random numbers. We use the *CPRNG Secrets*[2] python module to generate random numbers.

[2] https://docs.python.org/3/library/secrets.html.

Chapter 4

Introducing Structure in the Latent Space

4.1 CVAE with Information Factorization

We refer to our technique implementing the CVAE framework for anonymization as ObscureNet. ObscureNet is an encoder-decoder architecture augmented with as many discriminator networks as private attributes. Figure 4.1 shows the architecture of ObscureNet when there is only one private attribute associated with the sensor data that we wish to protect. ObscureNet conditions the decoder on the private attributes (similar to a CVAE) and performs adversarial information factorization to ensure that learned latent representations are invariant to private attributes. To this end, it utilizes discriminator networks trained using an adversarial method, each predicting the probability distribution over one private attribute given the latent variables.

We now describe how ObscureNet is trained given a dataset of samples with known private and public attributes and how the private attribute modification technique anonymizes the input data. This training can be done in a cloud server or on the IoT device if it has access to the public dataset used for training (i.e., $\mathcal{D}$). If the network is trained in the cloud, the weights and parameters of the encoder and decoder networks must be sent to IoT devices that run ObscureNet locally to anonymize their data.

4.1.1 Modification of Conditional Variables for data privacy

This section discusses how a CVAE model can be used to anonymize the private data attributes. Unlike the mean manipulation technique discussed in Chapter 3, the modification of the CVAEs conditional variable does not require calculation and propagation of the average values of latent representations to perform data anonymization.

Similar to Chapter 3 take into account the dataset $\mathcal{D} = \{(x_1, y_1, \bar{y}_1), ..., (x_m, y_m, \bar{y}_m)\}$ which is introduced in Section 2.8. Here, we discuss how by training a CVAE on this dataset and conditioning the decoder of the CVAE on the private attributes of the data, we can anonymize the reconstructed data through simple modification of the condition variable of the CVAE model.

Concretely, in the deployment phase, by encoding the input data embeddings, x_k through the CVAE, and conditioning the CVAE on the private attributes of data, y_k, we can manipulate the private attributes of the reconstructed data, $\tilde{x}_k$, through a simple modification of the condition variable. In this technique, by changing the condition of CVAE from y_k to y'_k, the private attribute of the reconstructed data ideally changes from y_k to y'_k.

Essentially, through the conditional variable of the CVAE, we are alleviating the issue of the VAE and the mean manipulation technique. In our discussion for the mean manipulation technique, we mentioned that due to our lack of ability in pinpointing specific latent variables corresponding to each private attribute, we perform the mean manipulation of all the latent variables. We are now using the private attribute input to the conditional variables of the CVAE to play the role of the deciding factor corresponding to the private attributes in the reconstructed data.

We can see that through the artificial injection of private attributes as input variables to the CVAE and changing those input variables, we can directly anonymize the data rather than using the mean manipulation technique.

4.1.2 The Need for Adversarial Training

Unfortunately, apart from the artificially injected private attributes as conditional variables to the CVAE, the original data and the rest of the latent variables generated by the encoder include information about the data's private attributes. Consequently, the leak of information about the private attributes through the rest of the latent variables undermines our ability to change the reconstructed data's private attributes.

This problem can be mitigated by enforcing and maximizing the disentanglement between latent variables and the condition variables. Fortunately, learning private attribute-invariant and attribute agnostic latent variables can be achieved by incorporating an adversarial objective into the CVAE's ELBO expressed in (2.2).

Through adversarial training, the networks can be trained to disentangle the condition from the latent representation [34]. We discuss this in the next section and see how we can modify the loss function to achieve this favorable disentanglement. Here, we train a neural network, Disc : $Z \to C$ that is trained in conjunction with the training of the CVAE. The neural network Disc is trained to infer the true condition corresponding to each latent representation z. Moreover, the encoder is trained to undermine the accuracy of the adversarial model by learning latent variables that capture the least possible amount of information about the condition.

4.1.3 Adversarial Training for Private Information Minimization

We first describe how the ELBO of a CVAE is modified for training an encoder-decoder architecture that learns a latent representation which contains little or no information about the private attributes. We then outline the process of training ObscureNet using an iterative minimax algorithm [34]. Without loss of generality and for ease of presentation, in the following, we consider the case that there is only one private attribute we want to conceal. Should there be more private attributes, the decoder must be conditioned on all these

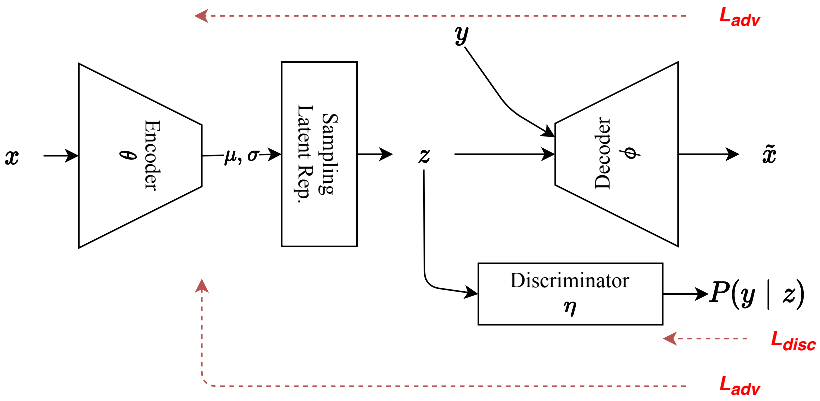

Figure 4.1: Training of ObscureNet

attributes and the adversarial loss function should include the cross-entropy loss of multiple discriminators.

Figure 4.1 shows the architecture of ObscureNet and the flow of gradients through the networks. In ObscureNet, the CVAE is augmented with a discriminator network which outputs $P_\eta(y|z)$, i.e., private attribute class-membership probabilities given the latent representation of input data. Here, η represents trainable parameters of the discriminator. If the private attribute and learned latent variables are completely disentangled, the discriminator would not be able to predict the private attribute.

The discriminator network can be trained using binary or categorical cross-entropy loss depending on whether the corresponding private attribute is binary (e.g., male or female) or categorical (e.g., weight). The loss of the discriminator network, $\mathcal{L}_{disc}$, can be expressed as follows:

$$\mathcal{L}_{disc}(\eta|\theta) = -\frac{1}{m}\sum_{(x,y)\in\mathcal{D}} \log P_\eta(y|z) = -\frac{1}{m}\sum_{(x,y)\in\mathcal{D}} \log P_\eta(y|G_\theta(x)), \tag{4.1}$$

where m is the number of samples in $\mathcal{D}$ and $G_\theta(x)$ is a function that compactly represents both encoding of x and sampling z from a multivariate Gaussian distribution. This is similar to the reparameterization trick used in [29].

The adversarial loss function can be written as:

$$
\begin{aligned}
\mathcal{L}_{adv}(\theta, \phi|\eta) = -\frac{1}{m} \sum_{(x,y)\in\mathcal{D}} \Bigg(& \mathbb{E}_{z\sim q_\theta(z|x)} \log p_\phi(x|z, y) \\
& - \beta \; D_{\mathrm{KL}}\big(q_\theta(z|x)||p(z)\big) \\
& - \alpha \; \log P_\eta(y|G_\theta(x))\Bigg)
\end{aligned} \qquad (4.2)
$$

It combines the discriminator loss (4.1) with CVAE's ELBO from Equation (2.2). We use the standard scalarization method and introduce weights which determine the relative importance of different terms in the adversarial loss function. The weights α and β are respectively assigned to the discriminator loss and the KL-divergence term. We treat these weights as hyperparameters and tune them in Chapter 5 to navigate the trade-off between utility and privacy.

ObscureNet is trained using an iterative algorithm, described in Algorithm 2. The discriminator is trained to predict y given z while the CVAE is trained to minimize the accuracy of the discriminator and minimize the loss function of the CVAE. While training the discriminator with the parameter, η, gradients are stopped from updating the CVAE network parameters: θ and ϕ. In the same respect, gradients are stopped from updating the discriminator network's parameter when training the CVAE.

Algorithm 2 shows the minibatch gradient descent for training ObscureNet, where B is the size of the minibatch. Both forward and the backward passes of ObscureNet's adversarial training can be seen in the pseudocode.

We make a critical remark that ObscureNet utilizes a different set of encoder and decoder networks for each public attribute. Each pair of these networks are trained separately using only samples in $\mathcal{D}$ that have the same public attribute. We argue that this helps to reduce the number of layers and neurons in the neural networks, making it easier to run ObscureNet on resource-constrained devices[1].

4.1.4 Anonymization with ObscureNet

After training the networks in an adversarial setting, we use them to perform anonymization before sharing sensor data with third-party applications

```
Algorithm 2: Training ObscureNet
  Data: Training dataset $\mathcal{D}$, learning rate $\lambda$, minibatch size $B$
  Result: Network parameters $\theta$, $\phi$, $\eta$
  $\theta, \phi, \eta \leftarrow$ initial values
  repeat
      Sample a minibatch: $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_B\}$
      /* pass samples through the networks */
      $\boldsymbol{\mu}, \boldsymbol{\sigma} \leftarrow \mathrm{Enc}(\boldsymbol{x}; \theta)$
      $\boldsymbol{\epsilon} \sim N(0, I)$
      $\boldsymbol{z} \leftarrow \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
      $\tilde{\boldsymbol{x}} \leftarrow \mathrm{Dec}(\boldsymbol{z}, y; \phi)$
      $P(y|\boldsymbol{z}) \leftarrow \mathrm{Disc}(\boldsymbol{z}; \eta)$
      Estimate gradients of minibatch
      /* Update parameters using gradients */
      $\eta \leftarrow \eta$ - $\lambda \nabla_{\eta}\ \mathcal{L}_{disc}(\eta|\theta)$
      $\{\theta, \phi\} \leftarrow \{\theta, \phi\}$ - $\lambda \nabla_{\{\theta,\phi\}}\ \mathcal{L}_{adv}(\theta, \phi|\eta)$
  until convergence of parameters ($\theta$, $\phi$, $\eta$)
```

that run locally or uploading it to cloud servers that host these applications. In particular, sensor data embeddings are passed through ObscureNet which obscures their private attributes, i.e., it generates a new version of each embedding with private attribute(s) that might be different from the original version. This process is depicted in Figure 4.2. Keep in mind that the encoder and decoder are the same networks trained using samples in $\mathcal{D}$. As it can be seen, in addition to encoder and decoder networks, we take advantage of a classification model that is trained separately to identify the public and private attributes associated with each sample in the test dataset. These attributes are denoted by $\hat{\hat{y}}$ and $\hat{y}$ respectively. The classification model will be needed as the true public and private attributes associated with the input data are not known at anonymization time. Identifying public attributes is necessary to select a CVAE network for ObscureNet, as discussed in the previous section.

Three different anonymization techniques can be implemented using ObscureNet. We refer to these techniques as *deterministic modification*, *probabilistic modification*, and *randomized* approach. They differ in whether they utilize the identified private attribute, and how they modify this attribute before it is used as a condition for the probabilistic decoder. We explain each of

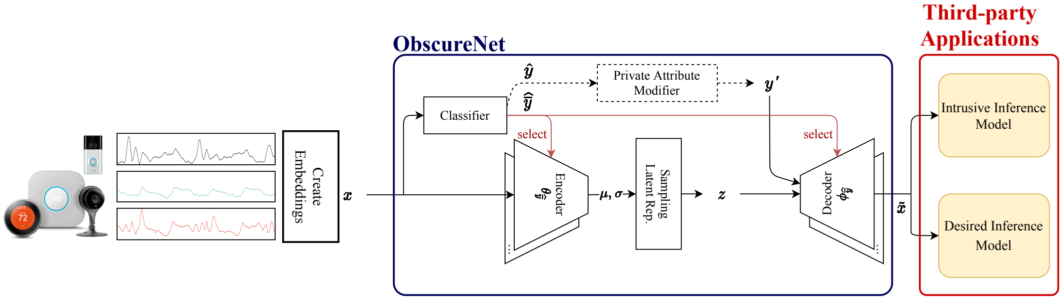

Figure 4.2: Anonymization with ObscureNet.

these approaches below.

Deterministic modification of the identified private attribute

This anonymization technique involves an injective function which deterministically maps each private attribute class in $\mathcal{A}$ to a different class in that set. This injective function is labelled as *private attribute modifier* in Figure 4.2. The identified private attribute (i.e., the output of the Classifier) is changed through the use of this modifier. We then pass the one-hot encoding of its output along with the latent representation of input data to the decoder, which produces a new version of the input data, denoted by $\tilde{x}$.

Probabilistic modification of the identified private attribute

Probabilistic modification is similar to the deterministic one with one exception: the mapping of private attributes is done probabilistically. That is, one of the K private attribute classes, $a_1, \cdots, a_K$, is picked at random for each sample in the test dataset, and the decoder is fed the one-hot encoding of this private attribute class along with the latent representation of data. The probabilistic modification is an effective defense against the user re-identification attack as we discuss in Chapter 5.

Randomized approach

The third anonymization technique eliminates the need for a classification model to identify the private attributes. Rather than identifying the private attribute first and modifying its one-hot encoding, it simply passes a *stochastic*

Algorithm 3: Sensor Data Anonymization w/ ObscureNet

```
Data: Data embedding $\boldsymbol{x}$, autoencoder parameters $\theta, \phi$
Result: Anonymized embedding $\tilde{\boldsymbol{x}}$
$\hat{y},\ \hat{\hat{y}} \leftarrow$ Classify($\boldsymbol{x}$)
$\boldsymbol{\mu}, \boldsymbol{\sigma} \leftarrow \mathrm{Enc}_{\hat{\hat{y}}}(\boldsymbol{x}; \theta)$
$\boldsymbol{\epsilon} \sim N(0, I)$
$\boldsymbol{z} \leftarrow \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
$y' \leftarrow$ Modify($\hat{y}$)                         // or $y' \leftarrow$ Randomize()
$\tilde{\boldsymbol{x}} \leftarrow \mathrm{Dec}_{\hat{\hat{y}}}(\boldsymbol{z}, y'; \phi)$
```

vector along with the latent representation of data to the decoder to produce a new version of this data. A stochastic vector is a vector of size K with non-negative entries that add up to 1. This technique aims to prevent user re-identification but is more straightforward than the probabilistic modification technique as it does not require training an additional classification model for the private attribute.

Algorithm 3 shows the steps of the deterministic and probabilistic modification techniques. Similar to the methodology used in Chapter 3, we use attribute-specific CVAEs with encoder and decoder networks tuned for each public attribute. We use the inferred public attribute values from the Classifier, $\hat{\hat{y}}$ to select the proper encoder and decoder networks. In the randomized approach, a randomly generated stochastic vector is used instead of the one-hot encoding of y'. In Chapter 5, we compare the three anonymization techniques presented above in terms of their ability to prevent user re-identification.

Chapter 5

Evaluation

5.1 Dataset

We use two open HAR datasets to evaluate the efficacy of our anonymization techniques in reducing the accuracy of intrusive inferences while maintaining the accuracy of desired inferences. We describe these two datasets (MotionSense and MobiAct) below and elaborate on the process of creating embeddings of time series sensor data.

5.1.1 MobiAct Dataset

The MobiAct [60] dataset is comprised of IMU readings from accelerometer and gyroscope sensors. The readings are collected from 66 subjects performing 12 different activities, including walking, running, climbing up and down the stairs. We consider data from a group of 37 subjects only to create a more balanced dataset that has roughly the same number of male and female subjects. Out of the 37 subjects we select, 17 are female, and the remaining 20 are male. Also, from the 12 different activities captured in the dataset, we consider the following 4 activities: walking, standing, jogging, and climbing up the stairs. We choose these activities for two reasons. First, these are the same activities captured in the MotionSense dataset, so we can compare the two datasets. Second, limiting our study to these activities partly addresses the class imbalance problem.

In our experiments, we assume that the activity exercised by a subject is the public attribute and must be inferred by a fitness tracking application.

However, the subject's weight and gender are deemed private, and their inference by this application or other third-party applications is regarded as a violation of user privacy. We model gender as a binary attribute since these are the two classes that are present in our dataset. We model weight as a ternary attribute using a simple binning strategy that tries to assign roughly the same number of subjects to each bin. This helps to address the class imbalance problem. In particular, subjects who weigh less than or equal to 70 kg are assigned to weight-group 0. Subjects who weigh between 70 and 90 kg are assigned to group 1, and the rest are assigned to weight-group 2.

5.1.2 MotionSense Dataset

The MotionSense [42] dataset is collected by accelerometer and gyroscope sensors of an iPhone 6s. The data obtained from accelerometer and gyroscope sensors has a sampling rate of 50 Hz. Each reading consists of 12 features, including attitude (roll, pitch, yaw), gravity, rotation rate, and user acceleration in three dimensions.

MotionSense contains data from 24 subjects (14 male and 10 female subjects). Each individual in this dataset performs 15 trials of 6 different activities. These activities include climbing up and down the stairs, walking, jogging, sitting, and standing. This dataset's subjects have a wide range of values for their age, weight, and height. Following [42] we combine the standing and sitting activities into one activity. This is done because the sensor reading is the same and distinguishing between these two activities is simple. Similar to the MobiAct dataset, we assume in our experiments that the activity exercised by a subject is the public attribute. We also assume that the gender identity of subjects is a private attribute and a third-party application should not be able to detect it. From all the 15 trials, we use trials 11, 12, 13, 14, 15, and 16 to build our test set. This is similar to the test set used in [42].

5.1.3 Embedding Sensor Data

It is shown in related work that activities can be identified more accurately if several consecutive IMU samples are analyzed at once. We call this an embedding of sensor data. To create our data embeddings, we use windows of size 128 samples. These windows are moved with strides of 10 samples to create the next embedding. For the MotionSense dataset, we combine features along the three axes to create one feature (i.e., the magnitude). However, for the MobiAct dataset, we use readings along the three axes as three separate features. Our experiments suggest that using three-dimensional sensor data increases the model accuracy for the MobiAct dataset. We use a trial-based partitioning of training and test data for MotionSense, and a partitioning for MobiAct dataset with 80% training set and 20% test set.

5.2 Comparison with Baselines

We compare ObscureNet, Chapter 4 and the augmented VAE network, Chapter 3 with baseline anonymization methods discussed below. Unless otherwise stated, ObscureNet is trained with hyperparameters that are set as follows: $\alpha = 0.2$ and $\beta = 2$. We consider the following baseline methods:

- 'General VAE' and 'Attribute-specific VAE' which rely on the mean manipulation technique explained in Section 3.1.1. The only distinction between these two methods is that the former, which is the method proposed in our previous work [22], uses a single VAE to anonymize all samples regardless of the value of their public attribute. The latter, however, trains separate VAEs for different public attributes. At anonymization time, it first detects the public attribute of input data and then chooses the appropriate VAE for learning and manipulating the latent representation.

- 'Attribute-specific CAE' and 'Attribute-specific CVAE' are conditional generative models where the condition represents the private attribute. Data is anonymized by altering the condition variable before sending it to the decoder as discussed in Section 4.1.1. The difference between these two baselines is that in the former, the condition is introduced in a vanilla autoencoder (resembling the architecture of Fader Networks, which are developed for manipulating images [34]), whereas in the latter, the condition is introduced in a variational autoencoder. Both methods train and utilize different autoencoders for different public attributes.

- ‘Anonymization Autoencoder (AAE)’ which is proposed in [42]. It does not use conditional generative models. Instead, it takes advantage of several regularizer models for adversarial training.

Moving from the top to the bottom of this list, the baseline methods combine different ideas to increase disentanglement of latent variables, making them more efficient and capable of concealing private attributes. Through ablations, we highlight the importance of incorporating each of these ideas in the design of ObscureNet, which essentially adds adversarial information factorization to the ‘Attribute-specific CVAE’ baseline and leverages non-deterministic private-attribute modifiers to prevent re-identification of private attributes after anonymization.

Accuracy of Sensitive and Desired Inferences

As the first step in our evaluation, we look at the accuracy of sensitive and desired inference models when their input is the original data and when it is the data anonymized by ObscureNet (Chapter 4), augmented VAE network (Chapter 3), and other baselines. We evaluate these methods in three different anonymization tasks: gender anonymization in MotionSense (*desired* = *activity*, *sensitive* = *gender*), gender anonymization in MobiAct (*desired* = *activity*, *sensitive* = *gender*), and finally weight-group anonymization in MobiAct (*desired* = *activity*, *sensitive* = *weight-group*). We only study the problem of hiding a single private attribute. An extension to the case where there are multiple private attributes to be obscured simultaneously is discussed in Section 5.4.2.

The results reported for anonymization in this section are obtained using deterministic private attribute modifier and mean manipulation techniques. We argue that if these techniques can reduce the accuracy of a sensitive inference to zero, we can achieve the accuracy of a random guess through randomization. This also prevents the re-identification of private attributes. Thus, we favor lower accuracy results for our deterministic anonymization results.

Table 5.1 shows the inference accuracy achieved when using the output of different baselines and our proposed methods in the MotionSense gender anonymization task. Moreover, the overall F1-scores for activity and gender inferences are given in the last two columns. The first row, labeled 'Original Data', indicates the accuracy of activity and gender inference models on the original (unanonymized) data. It can be readily seen that using attribute-specific VAEs improves the F1-score of activity inference from 65.51% obtained by a General VAE to 72.45%. This can be attributed to the fact that having a specific VAE for each public attribute can partly address the imbalance problem in the training data[1]. Unfortunately, this comes at the price of increasing the F1-score of gender inference. Comparing attribute-specific CAE and attribute-specific CVAE, we observe that both methods achieve comparable results for activity inference, but attribute-specific CVAE can effectively lower the gender inference accuracy and F1-score. We attribute this to the fact that variational autoencoders increase the latent variables' disentanglement and allow for straightforward generalization compared to vanilla autoencoders. We witness that the four baseline methods we discussed so far either fail to obscure the private data or significantly reduce its usefulness for desired inferences. We see that the augmented VAE network technique, Chapter 3 performs roughly similarly compared to the ObscureNet technique for the MotionSense dataset in anonymization of the gender attribute. However, as we see for the MobiAct dataset, Tables 5.2 and 5.3 indicate that the ObscureNet technique outperforms the augmented VAE network in gender and weight anonymization tasks.

The Replacement Autoencoder [43] cannot successfully obscure gender as

[1] It also reduces the size of the model. To illustrate this, we calculated the total size of the general VAE and multiple VAEs models. The general VAE model, for the MotionSense dataset, has 24.5 million trainable parameters. In the case of the MobiAct dataset, the general model has 8.7 million trainable parameters. Each of the attribute-specific VAEs has 1.7 million trainable parameters in the case of the MobiAct dataset (a total of roughly 7 million trainable parameters for all VAEs) and 0.5 million trainable parameters in the case of the MotionSense dataset (a total of 2 million trainable parameters for all VAEs).

Table 5.1: Gender anonymization results from the MotionSense dataset For each activity, the number of embeddings in the test set used for evaluation is shown in parentheses.

Method	Inference Accuracy								Accuracy/F1-score	
	Downstairs (1.9k)		Upstairs (2.5k)		Walking (6.2k)		Jogging (2.7k)		Overall	
	Activity	Gender	Activity	Gender	Activity	Gender	Activity	Gender	Activity	Gender
Original Data	95.58%	87.67%	93.19%	90.86%	98.71%	95.16%	97.28%	95.5%	96.93 / 95.90%	93.35 / 93.10%
General VAE	71.49%	38.62%	80.24%	28.34%	91.5%	45.12%	83.43%	38.61%	84.80 / 65.51%	39.72 / 38.05%
Attribute-specific VAEs	91.94%	76.22%	85.09%	64.05%	93.14%	77.9%	97.02%	23.12%	77.77 / 72.45%	63.94 / 61.95%
Attribute-specific CAEs	92.81%	63.74%	92.75%	77.60%	98.46%	76.13%	97.21%	85.52%	96.33 / 95.23%	76.54 / 74.81%
Attribute-specific CVAEs	92.09%	60.25%	93.31%	71.78%	96.13%	52.98%	96.8%	72.90%	95.39 / 94.05%	61.60 / 59.30%
AAE [42]	84.65%	57.91%	97.18%	57.64%	91.82%	52.89%	99.65%	46.23%	93.39 / 92.01%	53.15 / 42.83%
Augmented VAE network	89.73%	26.25%	93.47%	17.03%	98.49%	15.34%	97.28%	18.39%	96.04 / 94.76%	17.84 / 17.74%
ObscureNet	87.52%	27.48%	92.83%	19.0%	98.71%	15.94%	96.87%	10.5%	95.63 / 94.23%	17.06 / 16.34%

Table 5.2: Gender anonymization results from the MobiAct dataset. For each activity, the number of embeddings in the test set used for evaluation is shown in parentheses.

Method	Inference Accuracy								Accuracy/F1-score	
	Walking (42.9k)		Standing (43.2k)		Jogging (4.2k)		Upstairs (1k)		Overall	
	Activity	Gender	Activity	Gender	Activity	Gender	Activity	Gender	Activity	Gender
Original Data	98.09%	99.63%	99.53%	95.52%	99.78%	99.28%	95.45%	94.47%	98.82 / 91.46%	97.61 / 97.52%
General VAE	92.34%	88.53%	98.56%	63.83%	90.52%	87.07%	52.93%	66.47%	94.77 / 77.55%	76.54 / 76.49%
Attribute-specific VAEs	95.48%	90.06%	99.63%	52.73%	98.14%	93.36%	93.32%	82.68%	97.54 / 86.23%	72.53 / 75.63%
Attribute-specific CAEs	93.22%	25.06%	99.59%	80.89%	97.47%	75.68%	94.82%	78.45%	96.45 / 83.15%	54.39 / 54.37%
Attribute-specific CVAEs	92.66%	14.65%	99.65%	28.75%	96.55%	25.67%	94.3%	58.81%	96.16 / 81.86%	22.31 / 22.35%
AAE [42]	96.96%	58.13%	99.61%	42.12%	99.61%	56.72%	84.44%	58.60%	98.19 / 87.98%	50.49 / 45.73%
Augmented VAE network	87.57%	10.53%	99.67%	31.98%	96.75%	16.51%	94.63%	28.99%	93.79 / 78.51%	21.16 / 23.66%
ObscureNet	91.82%	5.39%	99.54%	23.48%	96.11%	10.55%	91.10%	24.46%	95.66 / 81.23%	14.39 / 14.28%

the average accuracy results of activity and gender inferences are 96.3% and 97.1%, respectively. AAE [42] significantly reduces the gender inference accuracy but cannot beat ObscureNet and the augmented VAE network. ObscureNet reduces the F1-score of gender inference by an additional 36%. This is done while achieving a comparable activity inference F1-score with the best baseline methods. It is worth mentioning that going downstairs is the most challenging activity to detect after concealing the gender attribute. Comparing ObscureNet with the technique proposed in [69], which reduces the gender inference accuracy to roughly 60% as reported by the authors, we can conclude that our anonymization technique is superior[2].

Next, we investigate gender anonymization results for the MobiAct dataset. Table 5.2 shows the accuracy of activity and gender inferences on the original

[2] We were unable to reproduce the results of [69] and did not find their code in a public repository. Hence, we cannot use it as a baseline in the MobiAct dataset.

Table 5.3: Weight-group anonymization results from the MobiAct dataset. For each activity, the number of embeddings in the test set used for evaluation is shown in parentheses.

Method	Inference Accuracy								Accuracy/F1-score	
	Walking (42.9k)		Standing (43.2k)		Jogging (4.2k)		Upstairs (1k)		Overall	
	Activity	Weight	Activity	Weight	Activity	Weight	Activity	Weight	Activity	Weight
Original Data	98.17%	97.42%	99.55%	85.85%	99.74%	93.36%	95.16%	76.69%	98.86 / 91.96%	91.53 / 91.95%
General VAE	81.88%	46.86%	90.34%	44.59%	54.73%	56.13%	18.47%	47.15%	83.94 / 59.84%	46.22 / 37.14%
Attribute-specific VAEs	92.83%	70.04%	99.66%	71.79%	97.7%	53.87%	93.33%	61.34%	96.29 / 81.79%	70.03 / 65.51%
Attribute-specific CAEs	94.23%	49.75%	99.65%	76.85%	97.82%	88.21%	94.87%	59.79%	96.97 / 84.22%	64.45 / 64.21%
Attribute-specific CVAEs	94.88%	26.59%	99.7%	19.37%	94.59%	53.55%	95.44%	51.41%	97.15 / 84.28%	24.69 / 21.46%
AAE [42]	97.39%	63.77%	99.35%	50.15%	98.91%	72.66%	90.91%	57.16%	98.32 / 88.50%	57.66 / 56.44%
Augmented VAE network	89.15%	22.41%	99.75%	37.93%	94.43%	31.25%	95.36%	41.69%	94.48 / 83.52%	30.37 / 25.47%
ObscureNet	94.22%	7.58%	99.59%	14.12%	96.40%	21.07%	91.37%	29.5%	96.83 / 83.40%	11.54 / 10.80%

data and the data anonymized by ObscureNet and other methods. The results are quite similar to the gender anonymization results from the MotionSense dataset. In this case, going upstairs is the most challenging activity to detect after concealing the gender attribute. Compared to AAE, ObscureNet can significantly decrease the accuracy and F1-score of gender inference (by more than 30%) with a small loss of data utility ($\sim 6\%$).

Lastly, we consider weight-group anonymization results from the MobiAct dataset. Recall that weight-group is a ternary private attribute; thus, this experiment is to check if ObscureNet can hide non-binary private attributes. It can be readily seen from Table 5.3 that ObscureNet outperforms all baselines in terms of the intrusive inference accuracy by a considerable margin. This is while it only reduces the data utility insignificantly, i.e., less than $\sim 6\%$ compared to AAE.

To conclude, our experiments show that ObscureNet outperforms the baselines and autoencoder-based anonymization techniques from related work in all three tasks, we studied in this section. Compared to the augmented VAE network technique introduced in Chapter 3, ObscureNet, which is based on the CVAE framework, performs comparably better in the private attribute anonymization. Hence, from now, we establish ObscureNet as our selected anonymization technique to perform further studies.

In the MobiAct dataset, it completely obscures gender and weight-group with a small loss of data utility. We believe that data utility can be further

Table 5.4: Accuracy of sensitive and desired inferences using ObscureNet with different private-attribute modifiers in the three anonymization tasks.

	Original data		Random guess	Deterministic		Probabilistic		Randomized	
	Activity	Private	Private	Activity	Private	Activity	Private	Activity	Private
MotionSense (Gender)	96.94	93.33	50.00	95.61	16.95	96.02	54.32	95.99	58.49
MobiAct (Gender)	98.82	97.52	50.00	95.72	14.43	97.02	52.70	96.26	54.20
MobiAct (Weight-group)	98.84	91.67	33.33	96.86	11.47	97.57	49.78	97.45	43.02

improved by using different hyperparameters as discussed in Section 5.4.1. In the next section, we examine the effects of different private-attribute modifiers and corroborate that ObscureNet can prevent re-identification of private attributes thanks to non-deterministic modifications of these attributes. This is a significant improvement over other autoencoder-based anonymization techniques [42], [43].

5.3 Non-deterministic Anonymization

In this section, we compare the three anonymization techniques which can be implemented using ObscureNet and were described in Section 4.1.4. Two of the three techniques, namely probabilistic modification and randomized approach, add randomness to the anonymization process. This can effectively prevent an adversary from passing a dataset with known private attributes through ObscureNet and training a model to recover the original data based on the anonymized data and true private attributes.

Table 5.4 shows the accuracy of desired and sensitive inferences when the private attribute (noted in parentheses) is obscured using ObscureNet. Expectedly, the deterministic modifier yields the lowest sensitive inference accuracy because, unlike the other two techniques, it modifies the private attribute at all times. However, as we discuss in the next section, private attributes can be easily re-identified due to this anonymization's deterministic nature. Results for the other two techniques are quite similar; they can reduce the intrusive inference accuracy to the level of a random guess. The randomized approach's nice property is that it does not need to use a model to detect the private attribute before modifying it. This makes it a suitable and more practical choice for anonymization on resource-constrained devices.

5.4 Re-identification Accuracy

In the previous section, the ability of ObscureNet to anonymize private data was evaluated in deterministic and non-deterministic cases. Although the deterministic private-attribute modifier does a better job of reducing the accuracy of sensitive inferences, we show that it cannot prevent the re-identification attack.

The re-identification attack exploits the deterministic nature of anonymization to foil the anonymization process [22]. Suppose 20% of the anonymized data is leaked to the attacker, i.e., they know the true private attribute associated with this data and can leverage this knowledge to train a model to re-identify the true attribute for the rest of the data. To get this 20%, we randomly choose 20% of the anonymized data and evaluate the accuracy of the re-identification attack. We do 20 independent runs and report the average and standard deviation of the accuracy of the re-identification model. Figure 5.1 illustrates that both ObscureNet and Anonymization Autoencoder [42] fail to completely ward off the re-identification attack due to the deterministic nature of anonymization they perform. However, from Figure 5.1, we can conclude that probabilistic modification and randomized approach can significantly reduce the accuracy of a re-identification model.

Figure 5.1 shows that the randomized approach which is easier to deploy than the probabilistic attribute modifier, has nearly the same performance as the probabilistic one in the gender anonymization task. But, in the weight-group anonymization task, it further reduces the re-identification accuracy by roughly 13%.

5.4.1 Investigating Utility-Privacy Trade-offs

In this section, we investigate how the anonymization performance of ObscureNet can be enhanced by adjusting the two hyperparameters, α and β, when networks are being trained. Furthermore, we explore if users can trade utility for privacy by adjusting the hyperparameters. For brevity, we only study the gender anonymization problem using the MotionSense dataset and

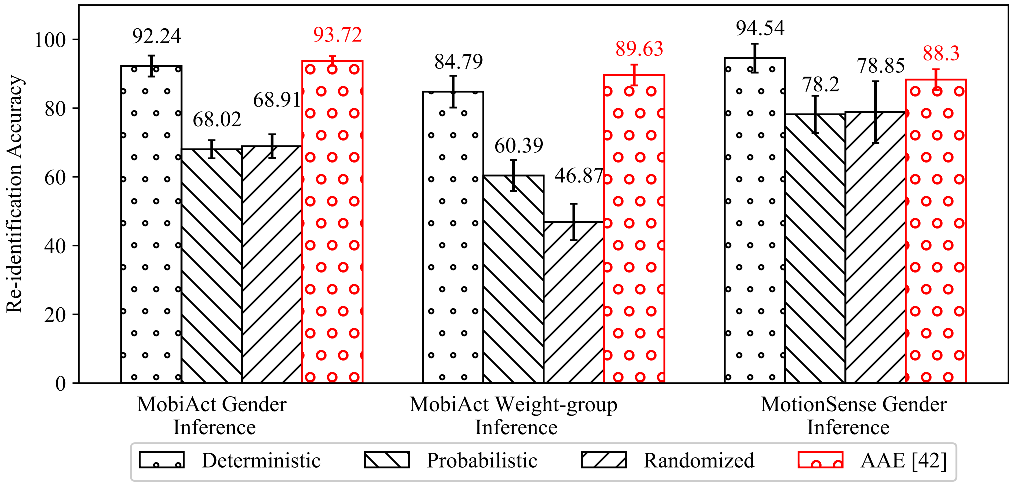

Figure 5.1: Comparison of the inference accuracy of re-identification model averaged over 20 runs on the sensor data anonymized by ObscureNet using different attribute modifiers. Error bars show 2σ from the mean.

report the results when the deterministic attribute modifier is adopted. Since we neglect the possibility of user re-identification in this stage, the best anonymization technique would be the one that reduces the accuracy of the sensitive inference to zero.

Recall the adversarial loss function of ObscureNet expressed in Equation (4.2). Intuitively, higher α encourages information factorization, which subsequently prevents the leak of private information through the latent representation. But it can lower the reconstruction quality because VAE's ELBO gets a lower relative importance. Similarly, higher β encourages the disentanglement of latent variables but reduces the reconstruction quality. Thus, it is possible to achieve different utility-privacy trade-offs by tuning α and β.

Figures 5.2 and 5.3 show respectively the accuracy of desired and sensitive inferences and how it changes with α and β values. We consider 6 values of β and 4 values of α; these values are intentionally chosen from the logarithmic scale to examine the range of behavior we can expect from ObscureNet. We assume β can take values from {0.1, 0.2, 0.5, 1, 2, 10} and α can take values from $\{0.1, 0.2, 1, 10\}$.

As it can be seen from Figure 5.2, for a fixed value of α, shifting β to the two extremes, i.e., 10 or 0.1, would diminish the utility of data, although the decline in utility is more pronounced when $\beta = 10$. Another observation is that for a fixed β, the value of α does not seem to drastically affect the utility of data unless it is equal to 10. We attribute this to the fact that when $\alpha = 10$, information factorization overwhelms the VAE's reconstruction loss.

In Figure 5.3, we can seen that increasing the value of β from 0.2 to 2 lowers the accuracy of the sensitive inference in general. However, moving β to any of the two extremes diminishes the anonymization performance of the ObscureNet for all α values. Comparing the curves for different values of α suggests that $\alpha = 0.1$ is almost always better than other values of α regardless of the value of β.

Considering the accuracy of both desired and sensitive inferences, it turns out setting α to 0.2 and changing β between 0.1 and 2 yields the Pareto frontier. For example, the user can trade utility for privacy by setting β to 2 and do the opposite by setting β to 0.1. While we only tried a small number of choices for α and β, we already showed that it is possible to navigate the privacy-utility trade-off by tuning these weights.

5.4.2 Obscuring Multiple Private Attributes

We now turn our attention to the case where there are multiple private attributes. Specifically, we treat gender and weight-group of subjects in the MobiAct dataset as private attributes and consider their activity as the public attribute. We evaluate the anonymization performance of a single ObscureNet model, which can hide both private attributes simultaneously, with deterministic, probabilistic, and randomized modifiers. In this case, ObscureNet conditions the probabilistic decoder on both private attributes and uses two discriminator networks, one for each private attribute. We train ObscureNet in an adversarial setting, as described in Section 4.1.3.

This problem is interesting as, in practice, the user often wishes to anonymize multiple private attributes simultaneously. Figure 5.4 shows the result of joint anonymization of gender and weight-group attributes while preserving information about the activity in the anonymized data. It is evident that Ob-

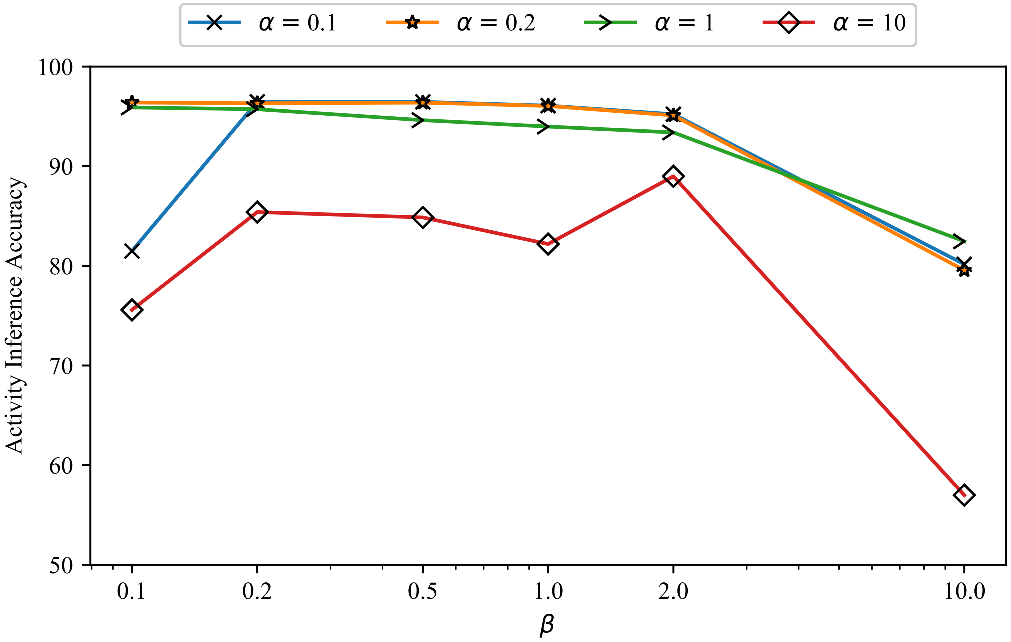

Figure 5.2: The accuracy of activity inference with varying α and β values. Note that the x-axis has logarithmic scale and the y-axis is exaggerated.

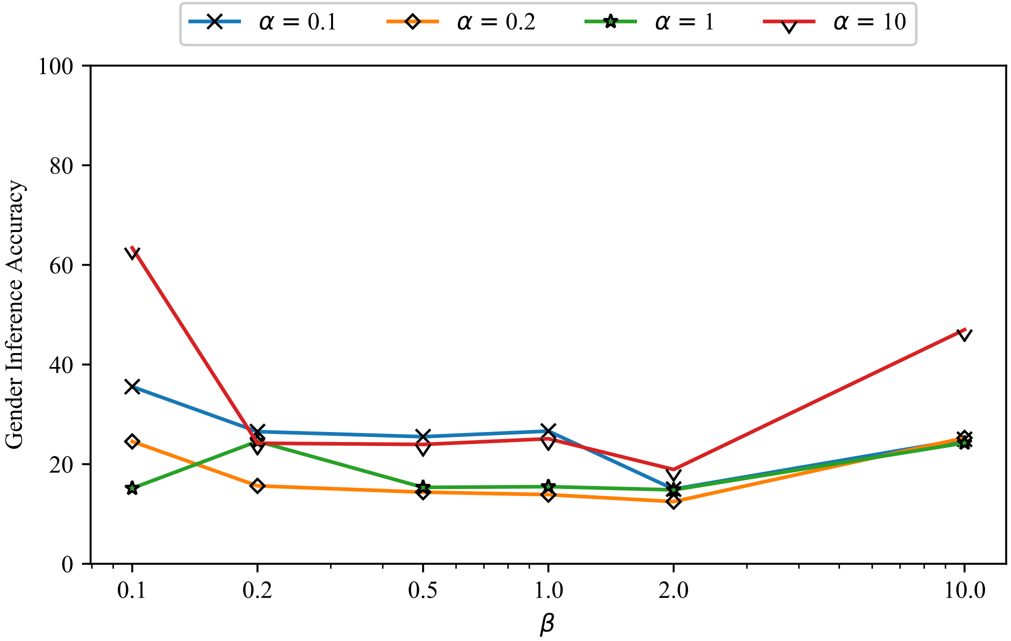

Figure 5.3: The accuracy of gender inference with varying α and β values. Note that the x-axis has logarithmic scale.

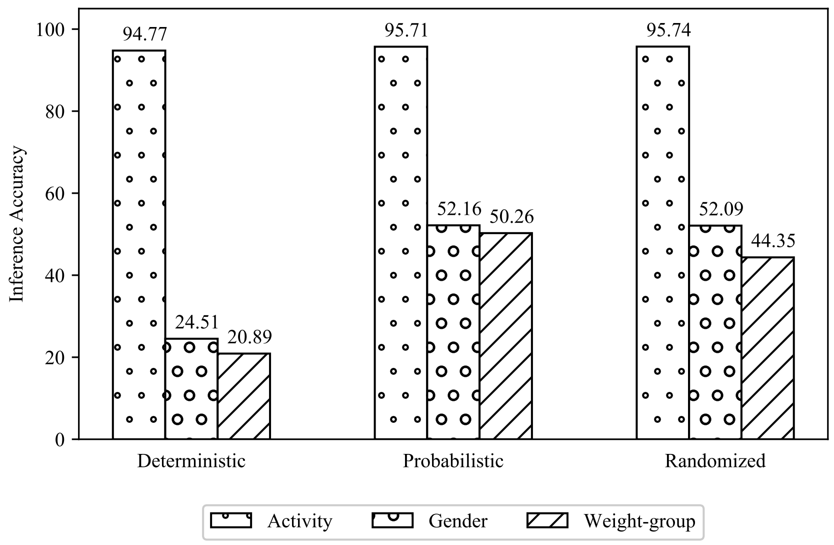

Figure 5.4: Inference accuracy of ObscureNet with different attribute modifiers when it is trained to hide two private attributes at the same time.

scureNet with a deterministic attribute modifier can successfully reduce the accuracy of both sensitive inferences to less than 25% while achieving the accuracy of 94.8% for the desired inference. Should we use the probabilistic modifier or the randomized approach, the accuracy of both sensitive inferences would get close to the level of a random guess. This indicates that a single ObscureNet model can effectively hide multiple private attributes.

Nevertheless, comparing this result with the result of removing each private attribute using a separate ObscureNet model (cf. Table 5.2 and Table 5.3), we can see that the anonymization performance is slightly degraded. Therefore, if the execution time and required resources are not an issue, an alternative approach for anonymizing several attributes could be to create an anonymization pipeline by utilizing multiple ObscureNet models (each concealing only one private attribute) and feeding the output of one model to the next model in the pipeline.

5.5 Performing Anonymization on IoT Devices

We finally investigate if ObscureNet can run on a Raspberry Pi 3 Model B to anonymize sensor data in real time. We use the Raspberry Pi as an IoT

Table 5.5: The running time of ObscureNet when anonymizing one embedding in different tasks

Anonymization task	Running time (ms)/Embedding				Total running time (ms) per embedding
	Desired Inference	Sensitive Inference	Probabilistic Encoder	Probabilistic Decoder	
MotionSense (Gender)	0.60	0.62	0.86	0.83	2.91
MobiAct (Gender)	10.39	10.14	9.33	1.15	31.01
MobiAct (Weight-group)	10.05	10.05	9.78	1.18	31.06

device that collects data from several sensors and runs ObscureNet locally to anonymize the collected data. We install Keras and PyTorch libraries on the Raspberry Pi, and report the running time of ObscureNet when the private attribute is modified in a probabilistic fashion[3]. We assume the encoder and decoder networks of ObscureNet are trained in a server, where the training data resides, and the weights are sent to the IoT device prior to anonymization.

To make real-time anonymization possible on the the Raspberry Pi, ObscureNet must be able to anonymize an embedding before the next one becomes available. Recall that in both datasets, we set the stride length to 10 samples, which means that the next embedding is created after receiving 10 sensor readings. The sampling rate of the IMU sensor is respectively 50Hz and 20Hz in MotionSense and MobiAct. Hence, a new embedding is created every 200 milliseconds in MotionSense, and every 500 milliseconds in MobiAct. If the running time of ObscureNet per embedding is less than this, it will be able to perform anonymization in real time.

As illustrated in Figure 4.2, an execution of ObscureNet can be divided into four main steps: (1) predicting the public and private attributes associated with the original data using the pre-trained desired and sensitive inference models, (2) encoding the input data through an attribute-specific encoder, (3) modifying the predicted private attribute, and (4) decoding the latent representation together with the modified attribute through an attribute-specific decoder. The first step involves running both inference models. The second and fourth steps require selecting the attribute-specific encoder and decoder networks according to the predicted public attribute. The third step involves generating a random number to determine how the private attribute should be

[3] the running time would be even lower if we adopt the randomized approach. This is because we do not need to predict the private attribute before modifying it.

modified. We ignore the running time of this step as it is negligible compared to the running time of the other three steps.

Table 5.5 shows the running time (in milliseconds) of the main steps in ObscureNet in three different anonymization tasks, namely gender anonymization in MotionSense, and gender and weight-group anonymization in MobiAct. To obtain the running time per embedding, we calculated the total running time of each step for approximately 8,000 embeddings and then divided this by the number of embeddings. Note that the running times of the attribute-specific encoder and decoder networks depend on the predicted public attribute. In this table, we only report the worst-case running times of the attribute-specific encoder and decoder networks across different activities (i.e., values of the public attribute).

Considering the gender anonymization task in MobiAct, the activity and gender inference models take roughly 10 milliseconds each to predict the private and public attributes of one embedding. The encoder and decoder running times for one input data embedding are around 9 and 1 milliseconds, respectively. These add up to 31 milliseconds per embedding. In the case of weight-group anonymization, the running times also add up to roughly 31 milliseconds. Given the time budget of 500 milliseconds, our results show that ObscureNet can anonymize the gender and weight-group attributes of participants in the MobiAct dataset in real time on a Raspberry Pi 3 model B.

Turning our attention to the gender anonymization task in MotionSense, we find that predicting the private and public attributes takes much less time. In particular, gender and activity inferences complete in 0.62 and 0.60 milliseconds, respectively. Moreover, the encoder and decoder networks take respectively 0.86 and 0.83 milliseconds to run. Thus, the total running time of ObscureNet would be 3 milliseconds per embedding. Given the time budget of 200 milliseconds, we corroborate that ObscureNet is capable of anonymizing input data embeddings of MotionSense in real time.

Chapter 6

Conclusion

In this chapter, we revisit the research questions outlined in Chapter 1 and explain how we addressed them in this thesis. We discuss the limitations of each of our proposed techniques and make some concluding remarks.

This thesis proposes anonymization solutions based on generative models which aim to offer acceptable levels of utility and privacy loss, while preventing user re-identification. Specifically, we proposed two techniques: an augmented VAE network described in Chapter 3 and a CVAE-based network called ObscureNet which was described in Chapter 4. To our knowledge, these ideas have not been previously applied to the sensor data anonymization problem. The proposed anonymization techniques are well-suited for deployment on resource-constrained edge devices.

In Chapter 3, we propose the augmented VAE network that utilizes our suggested mean manipulation framework for data anonymization [22]. The technique discussed in Chapter 3 is an extension of our work [23]. The mean manipulation technique uses a transformation function visualized in Figure 3.2. In Chapter 3, we augment the loss function of the VAE expressed in Equation (2.1) with a classification loss which results in the loss function in Equation (3.1).

Moreover, in Chapter 4 of this thesis, we propose a different take on the anonymization task by using CVAE-based conditional attribute modifications. We dub this technique as the ObscureNet anonymization technique. We discuss how the CVAE can be used along with the modification of its condition variables to change the private attributes in the reconstructed data. Here, we also, modify the original objective function of the CVAE, indicated in (2.2) by adding our proposed adversarial classification loss, shown in Equation (4.2). The goal here is that in order to increase the role of the condition variable in deciding the private data attributes, we minimize the amount of information about the private attributes in the learned latent representations of the CVAE model.

6.1 Addressing Research Questions

> *RQ 1. How successful are the two anonymization techniques considering data utility and privacy?*

Answer. Given our discussions in Chapter 3 and the results presented in Chapter 5, we conclude that compared to our attribute-specific VAE baseline, which uses the ELBO in Eq.(2.1), the modified VAEs' ELBO expressed in Eq.(3.1) proves to be more anonymization friendly and gives better deterministic anonymization results. We compare our anonymization baselines given deterministic manipulations of latent representations. This is because manipulation techniques that yield better sanitization of data in the deterministic case are shown to reduce inference accuracy to accuracy of random guess in the probabilistic case.

By examining the results from our evaluations, given in Chapter 5, it is evident that both the VAE and CVAE-based techniques proposed in this work outperform all the other baseline methods and the best-in-class techniques from related work. Upon further analysis of our techniques on three different anonymization tasks for the MotionSense and the MobiAct datasets, we see that ObscureNet outperforms our augmented VAE network anonymization method.

> *RQ 2. What results do these techniques yield in terms of anonymization performance and vulnerability to the re-identification attack when using non-deterministic manipulations?*

As discussed in Chapter 4, we can utilize multiple modification techniques while changing the private attributes of the data. We discussed deterministic, probabilistic, and randomized manipulations in Section 4.1.4. In our evaluation chapter, we study different characteristics of each of these modification techniques. As shown in Table 5.4, the deterministic modification renders the ObscureNet technique to be most effective in hiding the most amount of private attributes in the data. For example, in the anonymization of the gender attribute of the subjects in the MobiAct dataset, the deterministic anonymization reduces the gender inference accuracy from 97.52% to 14.43%. The probabilistic anonymization reduces the gender inference accuracy to only 52.70%. However, due to its deterministic nature, the deterministically modified data is vulnerable to the re-identification attack, as shown in Figure 5.1.

Moreover, in anonymization of the MobiAct gender private attributes, Figure 5.1 shows that the re-identification attack achieves a gender re-identification accuracy of 94.54% in the deterministic case. However, in the probabilistic anonymization, the gender re-identification accuracy achieves a comparably lower 78%. Hence, although the private attribute inference accuracy is reduced to a random guess accuracy for each case, we favor probabilistic and randomized methods because the anonymized data becomes less susceptible to the re-identification attack.

> *RQ 3. Can VAE and CVAE-based techniques be used to trade privacy for data utility?*

As discussed in Chapter 1, there is an inherent trade-off between utility of anonymized data and privacy. The utility-privacy trade-off indicates the trade-off between the amount of useful information removed in the process of concealing the private attributes of the data. The utility-privacy trade-off can be navigated by introducing training weights in the loss function of our proposed ObscureNet (refer to Eq. (4.2)).

Concretely, by introducing weights α and β in the loss function of our proposed ObscureNet, we can trade privacy for utility. As we see in Figures 5.2 and 5.3, changing the values of the α and β parameters forms a Pareto frontier for the utility-privacy trade-off. Hence, we can tune the anonymization process according to user needs and requirements.

> *RQ 4. Can our manipulation techniques be used to anonymize multiple private attributes all at once?*

In Chapter 5, we study how the ObscureNet technique can be utilized for the anonymization of two different private attributes at once, namely gender and weight-group attributes from the MobiAct dataset. These results are shown and presented in Figure 5.4, where the inference accuracy of both the gender and weight attributes are reduced to less than 25% while achieving an accuracy of 94.8% for the desired inference. However, by comparing these results with the result of removing each private attribute using a separate ObscureNet model, we can see that the anonymization performance is slightly degraded. Therefore, if the execution time and the required resources for anonymization are not our primary concerns, an alternative approach to anonymizing several attributes could be to create an anonymization pipeline by utilizing multiple ObscureNet models (each concealing only one private attribute) and feeding the output of one model to the next model in the pipeline.

6.2 Final Remarks

Given the rapid speed of adoption of consumer IoT devices, from indoor flying camera drones to smart vacuums and HVAC controllers, homes will soon be equipped with multiple sensing devices collecting information about their surroundings. Since the collected data can provide useful services, it is anticipated that consumers choose to buy many IoT devices for their convenience, essentially preferring convenience over their potential privacy risks. This indicates a need for privacy-preserving techniques that allow for safe and secure user data anonymization without limiting the penetration of IoT devices in the built environments.

Our experiments on two HAR datasets, MotionSense and MobiAct, suggest that among our two proposed methods, ObscureNet can reduce the accuracy of intrusive inferences by an additional 13.48% on average compared to the best autoencoder-based anonymization baseline without causing a significant drop in the accuracy of desired inferences. Compared to augmented VAE network, ObscureNet reduces the accuracy of intrusive inferences by 8.79% on average over the three anonymization tasks. In addition to giving better anonymization results, ObscureNet's modification of the condition variable is comparably easier and less costly than the mean manipulation technique used in the VAE anonymization. The need to calculate, update, and propagate new values of the average latent representations used in the mean manipulation process presents technical challenges for real-world deployment of this technique.

Furthermore, we demonstrate the ability of ObscureNet in concealing multiple private attributes at once. We investigate how by tuning hyperparameters in the training of ObscureNet, users can navigate the trade-off between utility and privacy. We argue that this is an essential property of our anonymization technique as users of IoT devices naturally have different expectations and concerns about applications they use, which work on their data.

In future work, We aim to provide users with abstract ways to describe their privacy concerns and add a slider knob to the anonymization technique to adjust their privacy in return for higher utility. Furthermore, We plan to study whether anonymization results vary with the subject/participant of the study. We will also look into ways to relax the assumption of having training data in a central repository and investigate how our VAE models can be trained in a federated learning setting.

References

[1] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[2] F. Alharbi, L. Ouarbya, and J. A. Ward, "Synthetic sensor data for human activity recognition," in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–9.

[3] K. Brkic *et al.*, "I know that person: Generative full body and face de-identification of people in images," in *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, 2017, pp. 1319–1328.

[4] B. Carbunar *et al.*, "Query privacy in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, pp. 1–34, 2010.

[5] Z. B. Celik *et al.*, "Sensitive information tracking in commodity iot," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1687–1704.

[6] M. Chanson *et al.*, "Blockchain for the iot: Privacy-preserving protection of sensor data," *Journal of the Association for Information Systems*, vol. 20, no. 9, pp. 1274–1309, 2019.

[7] Y. Chen, Y.-K. Lai, and Y.-J. Liu, "Cartoongan: Generative adversarial networks for photo cartoonization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9465–9474.

[8] C. De Canniere, O. Dunkelman, and M. Knezevic, "Katan and ktantan—a family of small and efficient hardware-oriented block ciphers," in *International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2009, pp. 272–288.

[9] F. Dernoncourt *et al.*, "De-identification of patient notes with recurrent neural networks," *Journal of the American Medical Informatics Association*, vol. 24, no. 3, pp. 596–606, 2017.

[10] A. Dorri *et al.*, "Blockchain in internet of things: Challenges and solutions," *arXiv preprint arXiv:1608.05187*, 2016.

[11] ——, “Blockchain for iot security and privacy: The case study of a smart home,” in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*, IEEE, 2017, pp. 618–623.

[12] ——, “Towards an optimized blockchain for iot,” in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, IEEE, 2017, pp. 173–178.

[13] C. Dwork *et al.*, “Calibrating noise to sensitivity in private data analysis,” in *Theory of cryptography conference*, Springer, 2006, pp. 265–284.

[14] ——, “The algorithmic foundations of differential privacy.,” *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.

[15] K. Fan *et al.*, “Lightweight rfid protocol for medical privacy protection in iot,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1656–1665, 2018.

[16] E. Fernandes *et al.*, “Flowfence: Practical data protection for emerging iot application frameworks,” in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 531–548.

[17] M. Fredrikson, S. Jha, and T. Ristenpart, “Model inversion attacks that exploit confidence information and basic countermeasures,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1322–1333.

[18] C. Gartenberg, *The google home mini’s mute switch makes privacy deliberate*, `https://www.theverge.com/circuitbreaker/2019/8/23/20828854/google-home-mini-mute-switch-button-privacy-microphones`, 2019.

[19] C. Gentry and D. Boneh, *A fully homomorphic encryption scheme*, 9. Stanford university Stanford, 2009, vol. 20.

[20] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[21] Z. Gu *et al.*, “Yerbabuena: Securing deep learning inference data via enclave-based ternary model partitioning,” *arXiv preprint arXiv:1807.00969*, 2018.

[22] O. Hajihassani, O. Ardakanian, and H. Khazaei, “Latent representation learning and manipulation for privacy-preserving sensor data analytics,” in *IEEE Second Workshop on Machine Learning on Edge in Sensor Systems (SenSys-ML)*, 2020, pp. 7–12.

[23] O. Hajihassani, O. Ardakanian, and H. Khazaei, *Privacy-preserving data analysis through representation learning and transformation*, 2020. arXiv: `2011.08315 [cs.LG]`.

[24] L. Hanzlik *et al.*, “Mlcapsule: Guarded offline deployment of machine learning as a service,” *arXiv preprint arXiv:1808.00590*, 2018.

[25] I. Higgins *et al.*, “β-VAE: Learning basic visual concepts with a constrained variational framework,” *ICLR*, vol. 2, no. 5, p. 6, 2017.

[26] C. Huang *et al.*, “Context-aware generative adversarial privacy,” *Entropy*, vol. 19, no. 12, p. 656, 2017.

[27] R. Jia *et al.*, “Privacy-enhanced architecture for occupancy-based hvac control,” in *2017 ACM/IEEE 8th international conference on cyber-physical systems (ICCPS)*, IEEE, 2017, pp. 177–186.

[28] L. Jiang *et al.*, “Differentially private collaborative learning for the iot edge.,” in *EWSN*, 2019, pp. 341–346.

[29] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[30] D. P. Kingma *et al.*, “Semi-supervised learning with deep generative models,” *Advances in neural information processing systems*, vol. 27, pp. 3581–3589, 2014.

[31] S. Kitchener, “Occupeye sensors: A sinister exercise in big brother-style management or a 21st-century way to monitor workers’ needs?” *Independent*, 2016. [Online]. Available: `https://www.independent.co.uk/news/media/occupeye-sensors-sinister-exercise-big-brother-style-management-or-21st-century-way-monitor-workers-needs-a6808281.html`.

[32] J. Klys, J. Snell, and R. Zemel, “Learning latent subspaces in variational autoencoders,” in *Advances in Neural Information Processing Systems*, ser. NIPS’18, 2018, pp. 6444–6454.

[33] J. Konecy *et al.*, “Federated learning: Strategies for improving communication efficiency,” *arXiv preprint arXiv:1610.05492*, 2016.

[34] G. Lample *et al.*, “Fader networks: Manipulating images by sliding attributes,” in *Advances in neural information processing systems*, 2017, pp. 5967–5976.

[35] M. Lentz *et al.*, “Secloak: Arm trustzone-based mobile peripheral control,” in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 1–13.

[36] N. Li *et al.*, “T-closeness: Privacy beyond k-anonymity and l-diversity,” in *2007 IEEE 23rd International Conference on Data Engineering*, IEEE, 2007, pp. 106–115.

[37] X. Li, J. Luo, and R. Younes, "Activitygan: Generative adversarial networks for data augmentation in sensor-based human activity recognition," in *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, 2020, pp. 249–254.

[38] Z. Lin *et al.*, "Using GANs for sharing networked time series data: Challenges, initial promise, and open questions," in *Proceedings of the ACM Internet Measurement Conference*, ACM, 2020, pp. 464–483.

[39] F. Liu and T. Li, "A clustering-anonymity privacy-preserving method for wearable iot devices," *Security and Communication Networks*, vol. 2018, 2018.

[40] Y. Lu *et al.*, "Blockchain and federated learning for privacy-preserved data sharing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2019.

[41] A. Machanavajjhala *et al.*, "L-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, 3–es, 2007.

[42] M. Malekzadeh *et al.*, "Mobile sensor data anonymization," in *Proceedings of the International Conference on Internet of Things Design and Implementation*, 2019, pp. 49–58.

[43] M. Malekzadeh, R. G. Clegg, and H. Haddadi, "Replacement autoencoder: A privacy-preserving algorithm for sensory data analysis," in *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation*, 2018, pp. 165–176.

[44] N. Malkin *et al.*, "Privacy attitudes of smart speaker users," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 4, pp. 250–271, 2019.

[45] F. McKeen *et al.*, "Intel software guard extensions (intel sgx) support for dynamic memory management inside an enclave," in *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016*, 2016, pp. 1–9.

[46] F. Mo *et al.*, "Darknetz: Towards model privacy at the edge using trusted execution environments," *arXiv preprint arXiv:2004.05703*, 2020.

[47] J. Ngiam *et al.*, "Multimodal deep learning," in *ICML*, 2011.

[48] D. Nie *et al.*, "Medical image synthesis with context-aware generative adversarial networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2017, pp. 417–425.

[49] S. A. Osia *et al.*, "A hybrid deep learning architecture for privacy-preserving mobile analytics," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4505–4518, 2020.

[50] H. Park *et al.*, "Streambox-tz: Secure stream analytics at the edge with trustzone," in *2019 {USENIX} Annual Technical Conference ({USENIX}{ATC} 19)*, 2019, pp. 537–554.

[51] P. Peris-Lopez *et al.*, "Advances in ultralightweight cryptography for low-cost rfid tags: Gossamer protocol," in *International Workshop on Information Security Applications*, Springer, 2008, pp. 56–68.

[52] Y. Rahulamathavan *et al.*, "Privacy-preserving blockchain based iot ecosystem using attribute-based encryption," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, IEEE, 2017, pp. 1–6.

[53] R. Shokri *et al.*, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2017, pp. 3–18.

[54] R. P. Singh *et al.*, "Tussleos: Managing privacy versus functionality trade-offs on iot devices," *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 3, pp. 1–8, 2018.

[55] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in neural information processing systems*, 2015, pp. 3483–3491.

[56] T. Song *et al.*, "A privacy preserving communication protocol for iot applications in smart homes," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1844–1852, 2017.

[57] K. Sun, C. Chen, and X. Zhang, """ alexa, stop spying on me!" speech privacy protection against voice assistants," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 298–311.

[58] L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[59] H. Tao *et al.*, "Secured data collection with hardware-based ciphers for iot-based healthcare," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 410–420, 2018.

[60] G. Vavoulas *et al.*, "The mobiact dataset: Recognition of activities of daily living using smartphones.," in *ICT4AgeingWell*, 2016, pp. 143–151.

[61] J. Wang *et al.*, "Sensorygans: An effective generative adversarial framework for sensor-based human activity recognition," in *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2018, pp. 1–8.

[62] Y. Wu *et al.*, "Privacy-protective-gan for privacy preserving face de-identification," *Journal of Computer Science and Technology*, vol. 34, no. 1, pp. 47–60, 2019.

[63] P. Xie *et al.*, “Crypto-nets: Neural networks over encrypted data,” *arXiv preprint arXiv:1412.6181*, 2014.

[64] K. Yang *et al.*, “Hardware designs for security in ultra-low-power iot systems: An overview and survey,” *IEEE Micro*, vol. 37, no. 6, pp. 72–89, 2017.

[65] S. Yao *et al.*, “Deep compressive offloading: Speeding up neural network inference by trading edge computation for network latency,” in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, ACM, 2020, pp. 476–488, ISBN: 9781450375900.

[66] Y. Yao *et al.*, “Defending my castle: A co-design study of privacy mechanisms for smart homes,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[67] K. Ying *et al.*, “Truz-droid: Integrating trustzone with mobile operating system,” in *Proceedings of the 16th annual international conference on mobile systems, applications, and services*, 2018, pp. 14–27.

[68] J. Yu *et al.*, “Generative image inpainting with contextual attention,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5505–5514.

[69] D. Zhang *et al.*, “Collective protection: Preventing sensitive inferences via integrative transformation,” in *2019 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2019, pp. 1498–1503.

[70] Y. Zhao *et al.*, “Privacy-preserving blockchain-based federated learning for iot devices,” *IEEE Internet of Things Journal*, 2020.

[71] S. Zheng *et al.*, “User perceptions of smart home iot privacy,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, no. CSCW, pp. 1–20, 2018.

[72] Y. Zhou *et al.*, “Auto-conditioned recurrent networks for extended complex human motion synthesis,” in *International Conference on Learning Representations*, 2018.

[73] D. Zhu *et al.*, “Tainteraser: Protecting sensitive data leaks using application-level taint tracking,” *ACM SIGOPS Operating Systems Review*, vol. 45, no. 1, pp. 142–154, 2011.

[74] S. Zhu *et al.*, “Automating visual privacy protection using a smart led,” in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017, pp. 329–342.