Privacy through Diffusion: A White-listing Approach to Sensor Data Anonymization

Xin Yang
University of Alberta
Edmonton, AB, Canada
xin.yang@ualberta.ca

Omid Ardakanian
University of Alberta
Edmonton, AB, Canada
ardakanian@ualberta.ca

ABSTRACT

Generative models have shown great promise in synthesizing high-quality time-series data that resemble the sensor data generated by mobile and IoT devices, but do not reveal the user's private attributes. These synthesized data can be treated as the obfuscated version of the sensor data and sent to downstream applications. However, existing obfuscation techniques that rely on generative models require the user to enumerate all inferences they deem intrusive. This black-listing approach would inevitably result in privacy loss if the definition of intrusive inferences changes after releasing the obfuscated data. In this work, we propose a white-listed approach to sensor data obfuscation based on a guided denoising diffusion model and a surrogate model for the desired inference. We evaluate this obfuscation model on a human activity recognition dataset and show that the proposed obfuscation model provides an acceptable privacy-utility trade-off, without assuming knowledge of the private attributes.

CCS CONCEPTS

• **Computer systems organization → Sensor networks**; • **Security and privacy → Privacy protections**.

KEYWORDS

Privacy-utility trade-off, deep generative models

ACM Reference Format:
Xin Yang and Omid Ardakanian. 2023. Privacy through Diffusion: A White-listing Approach to Sensor Data Anonymization. In *Proceedings of the 4th Workshop on CPS&IoT Security and Privacy (CPSIoTSec '23), November 26, 2023, Copenhagen, Denmark.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3605758.3623496

1 INTRODUCTION

Recent years have witnessed an increase in the number of mobile IoT devices equipped with a wide range of sensors. The rich data captured by these sensors are often shared with third-party applications and cloud service providers for enhanced user experience, personalized services, or storage. While the service provider is generally trusted to faithfully perform the desired computation, it may simultaneously perform intrusive inferences on this data, for example, to extract and monetize users' private information. Sharing raw sensor data with this *honest-but-curious* (HBC) server poses significant privacy risks to users. For example, several studies have shown that motion data collected by sensors in smartphones and wearables can be used to infer sensitive information about the user, such as their gender, age, and weight [6, 15]. This calls for designing effective privacy-preserving methods that can be deployed on mobile IoT devices to obfuscate sensor data before they are released to a third-party application or service provider (e.g., a fitness tracking app running on the mobile device or in the cloud).

Anonymization of sensor data streams gives rise to several important challenges. First, sensors do not directly output specific attributes of a user. They rather generate time-series data from which these attributes might be inferred using appropriate signal processing or machine learning techniques. As a result, privacy-preserving techniques, such as differential privacy (DP), cannot be readily applied to sensor data streams to limit the disclosure of private attributes that are not explicitly included in the dataset, i.e., DP noise must be added to private attribute information contained in sensor data without affecting public attribute information [23]. Moreover, obfuscation techniques must be computationally efficient; otherwise, they cannot run on battery-powered mobile IoT devices in real-time. Thus, secure multi-party computation based techniques are not well-suited for this application. Finally, the time-series data usually contain patterns that correlate with both private (e.g., age) and non-private (e.g., hand gesture) attributes of the user. The entanglement between private and non-private attributes leads to a trade-off between data utility and privacy loss because manipulating these patterns would reduce the accuracy of desired and unwanted inferences at once.

To address these challenges and provide a reasonable trade-off between utility and privacy of sensor data streams, various machine learning techniques have been used in prior work. In particular, adversarial machine learning has been used to collaboratively train multiple neural networks to extract privacy-preserving feature representations from sensor data [12–14]. The downside of this approach is that it requires a complete redesign of existing applications so that they ingest the extracted features instead of raw sensor data. In another line of work, data obfuscation techniques have been developed by taking advantage of an encoder-decoder architecture. For example, anonymization autoencoder (AAE) [15], ObscureNet [7], and Olympus [19] utilize autoencoders trained in an adversarial fashion to generate novel time-series data that can be used in desired inferences but deteriorate the accuracy of unwanted inferences. To train these architectures, users must specify one or more attributes that they wish to detect by sending their data to the

CPSIoTSec '23, November 26, 2023, Copenhagen, Denmark

ACM ISBN 979-8-4007-0254-9/23/11…$15.00
https://doi.org/10.1145/3605758.3623496

third-party service provider, in addition to one or more attributes that they wish to conceal from that service provider. We call the former type of attributes *public (non-private) attributes* and the latter ones *private attributes*. In the fitness tracking example, activity is the public attribute and gender, age, and weight may be listed as private attributes. While these obfuscation networks generate time-series data in the original input space and therefore can be readily consumed by existing applications, users are expected to specify all private attributes they wish to conceal, in addition to the public attribute(s). However, the definition of private attributes could change over time, and users cannot always foresee these changes. Hence with this *black-listing* approach, it is impossible to control the leakage of information about a new set of private attributes once the obfuscated sensor data are released.

We propose a *white-listing* approach to sensor data obfuscation, requiring users to specify *only* the public attribute(s) they wish to detect by using a third-party application or service.[1] The design of this obfuscation model is inspired by the denoising diffusion model [5, 8, 9, 21] – a generative AI model that achieves superb performance in the image synthesis task and exhibits better training stability than the generative adversarial network (GAN). In the forward diffusion process, a small amount of Gaussian noise is added to the sensor data in each timestep, so after a sufficiently large number of timesteps, the sensor data becomes identical to isotropic Gaussian noise. This process can be reversed by training a machine learning model to predict the noise introduced in each timestep, making it possible to synthesize sensor data from randomly sampled noise, which is the backward diffusion process. Our intuition is that by controlling the backward diffusion process, we can condition the diffusion model and guide it toward generating a copy of sensor data that contains information about the same public attribute as the original data, thereby ensuring high data utility. Since we do not impose any constraint on other attributes, they will be randomly sampled from the underlying distribution in the training set. So long as the distribution of other attributes is diversified in the training set, the synthesized data does not strongly correlate to a specific private attribute class and an HBC adversary can achieve an intrusive inference accuracy near the random guessing level. Our contributions are as follows:

- We design an obfuscation model using a diffusion model conditioned with latent representations that contain information about the public attribute(s). These representations are extracted from the original sensor data using a pretrained surrogate utility model. We show that a simple surrogate model can aid in generating obfuscated data that will be consumed by potentially more sophisticated utility models.
- We evaluate the proposed obfuscation model on a human activity recognition dataset and compare it with two obfuscation baselines that black-list a few user-specified private attributes. Our model maintains high data utility and achieves competitive privacy-preserving performance compared to the baselines, without having a priori knowledge of the private attributes.
- We outline multiple open research challenges pertaining to white-listed sensor data obfuscation and potential solutions that must be investigated in future work.

To our knowledge, this paper is the first to explore the feasibility of obfuscating sensor data using a guided diffusion model. The proposed white-listing approach elevates the practicality of black-listed obfuscation by eliminating the need for specifying private attributes and enabling future-proof privacy protection.

2 RELATED WORK

Privacy-aware feature extraction. On-device sensor data obfuscation techniques have garnered considerable attention owing to the recent progress in deep learning and the growing computing power of mobile IoT devices. A notable class of obfuscation techniques extracts a small number of features from sensor data such that these features do not contain sensitive information, thus they can be released in lieu of the original sensor data. Liu *et al.* [14] propose Privacy Adversarial Network (PAN) that utilizes an encoder to extract feature representations that do not contain private information through adversarial training. Li *et al.* propose DeepObfuscator [13] by training a convolutional neural network (CNN) comprising a feature extractor and the desired task classifier, together with two adversarial network components designed to reconstruct data and predict private attributes. TIPRDC [12] trains a feature extractor through an adversarial game to conceal private attributes. It maintains data utility by maximizing the mutual information between raw data and the combination of the private attribute and extracted feature. Since these works focus on extracting privacy-preserving feature representations, developers must update their application for it to be compatible with the obfuscated features.

Obfuscation through the lens of generative AI. Another line of work generates novel sensor data in the same space as the original sensor data by obfuscating sensitive information that it contains. Malekzadeh *et al.* [15] propose the use of an autoencoder-based architecture and multiple regularizers to conceal private attributes in the latent space of an autoencoder. Hajihassani *et al.* [7] propose ObscureNet that collaboratively trains a conditional variational autoencoder and an auxiliary network in an adversarial fashion to obscure private information in the latent space. Yang *et al.* [24] explain how a similar conditional variational autoencoder can be trained on decentralized data, possibly owned by many clients, to offer privacy protection during the generative model training process. Olympus [19] trains an autoencoder that optimizes both utility loss and inference accuracy to obfuscate data while maintaining its utility. These models retain the original data format, allowing existing applications to seamlessly consume the obfuscated data. However, these techniques require users to define a set of private attributes they wish to protect, which is equivalent to black-listing the respective inferences. Hence, the data obfuscation model trained by these approaches can hardly be extended to protect other attributes that users might consider private at a later time. On the contrary, our work explores a *white-listed* data obfuscation technique grounded on the diffusion process, which not only preserves the original data format for maximum compatibility with legacy applications, but also offers privacy protection for nearly all attributes apart from the ones considered public.

[1] Note the definition of the public attribute(s) remains unchanged as long as sensor data are sent to the same application. For example, in the context of fitness tracking, human activity is the public attribute naturally.

Guiding diffusion models for data synthesis. Recently, generative models based on the diffusion process [8, 11, 20] have gained tremendous attention for their superior performance in generating high-quality, novel images, and a more stable training process than GAN-based models. Ho *et al.* model the diffusion process as a Markov chain and propose Denoising Diffusion Probabilistic Models (DDPM) [8]. To accelerate the synthesis process, Song *et al.* [21] propose Denoising Diffusion Implicit Models (DDIM) that model the diffusion process as a non-Markov process, enabling one-step computation for any timestamp. To further improve the quality and control the content of the synthesized data, different approaches to guide the data generation process have been proposed in the literature [5, 9, 16]. Dhariwal *et al.* [5] introduce the notion of classifier-guided conditioning, which uses the gradients of an auxiliary classifier to condition the diffusion model on the class information. However, training such an auxiliary model under noisy conditions can be challenging and computationally intensive. Ho *et al.* [9] propose classifier-free conditioning, which directly conditions the diffusion model using the class label. Similarly, Preechakul *et al.* [17] propose using the latent representations learned from a semantic encoder to condition a DDIM. Other researchers have been utilizing pre-trained image-language models, such as Contrastive Language-Image Pre-Training (CLIP) [18], to guide the image generation process using text prompts [10, 16].

Unlike the classifier-guided diffusion model that utilizes the gradients computed by a classifier [5], our generative model takes advantage of the latent representation extracted by a pretrained auxiliary classifier to condition the backward diffusion process. It also differs from the classifier-free guidance approach [9] in two ways. First, in classifier-free guidance, only class labels are used as the condition variable. This reduces the dependence of the synthesized data on the original data and limits its ability to generate time-series that resemble sensor data that contains information about the same public attribute. Second, Ho *et al.* [9] simultaneously train a diffusion model with and without the condition, and use a convex combination of the output of the two models to control the diversity and quality of the synthesized data samples. In our implementation, we train the conditioned diffusion model only to avoid introducing excessive noise to the synthesized data.

3 DIFFUSION-BASED DATA OBFUSCATION

We present the architecture of our diffusion-based obfuscation model. The model is comprised of a denoising diffusion model and one (or multiple) auxiliary module(s). The denoising diffusion model learns to synthesize sensor data samples (i.e. time-series segments) by gradually denoising a randomly sampled Gaussian noise. To enable white-listed data obfuscation, we guide the synthesis process using only conditions that correspond to the user-specified public attribute(s). We assume the obfuscation model is trained on data gathered from a sufficiently diverse population, therefore any unspecified attribute in the synthesized data can be considered randomly sampled from the underlying distribution in the training set. For each public attribute white-listed by the user, we adopt an auxiliary classifier that acts as a surrogate of the model used in the third-party application that will consume the obfuscated data to predict the public attribute. The model adopted by third-party applications is presumably more accurate than the auxiliary classifier we use to train the diffusion model. We refer to this auxiliary classifier as the *surrogate utility model.* The first layers of the surrogate utility model (i.e. the encoder module) compresses the original sensor data into a latent representation that corresponds to the public attribute. This representation helps the diffusion model properly denoise and synthesize a sensor data sample that contains information about the same public attribute as the original data sample.

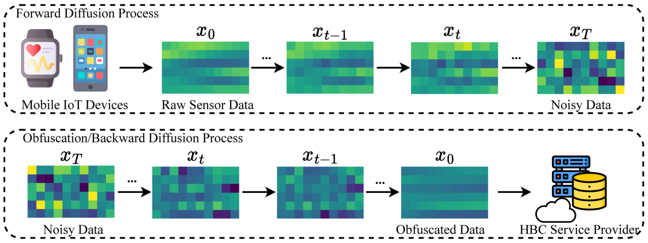


Figure 1: Diffusion process of the white-listed data obfuscation. The first 2D array (x_0) is the original multi-channel sensor data segmented using a fixed size window, i.e. rows are sensor channels and columns are time instants.

3.1 Conditional Denoising Diffusion Model

We build our conditional denoising diffusion model based on the recently proposed diffusion model with classifier-free guidance [9]. Specifically, for a forward diffusion process of T timesteps, a small amount of Gaussian noise is sampled and added to the input sensor data at every timestep. We denote the raw sensor data as x_0 and its distribution as $x_0 \sim q(x_0)$. The amount of added noise is modeled using a variance scheduler β, where $0 < \beta_1 < \beta_2 < \ldots < \beta_T < 1$. For simplicity, a linear scheduler [8] that starts from 0.0001 and ends at 0.02 with $T = 1000$ is used in our implementation. We model the diffusion process as a Markov process, i.e., DDPM. The noisy data at timestep t of the forward diffusion process can be written as:

$$p(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t \mathrm{I}). \tag{1}$$

However, using the above formula, the data at timestep t strictly depends on the previous timestep $t-1$, hence requiring precomputing and storing the data at all time steps. To address this issue, Equation 1 can be written as depend only on the input x_0 by introducing $\alpha_t = 1 - \beta_t$ and $\bar{\alpha} = \prod_{t=1}^{T} \alpha_t$:

$$p(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}}x_0, (1-\bar{\alpha})\mathrm{I}). \tag{2}$$

By further applying the reparameterization, we can write x_t as a closed-form expression w.r.t. x_0 and ϵ as:

$$x_t = \sqrt{\bar{\alpha}}x_0 + \sqrt{1-\bar{\alpha}}\epsilon, \tag{3}$$

where ϵ follows a Gaussian distribution $\epsilon \sim \mathcal{N}(0, \mathrm{I})$. Thus, for sufficiently large T, x_T will be nearly an isotropic Gaussian.

The intuition of the diffusion model is that, by reversing the forward diffusion process, one can reconstruct the raw data x_0 by sampling x_T from a Gaussian distribution $x_T \sim \mathcal{N}(0, \mathrm{I})$. The goal of $q(x_{t-1}|x_t)$ needs to be estimated using a neural network model p_θ considering it is intractable. When the noise schedule β is sufficiently small, $p_\theta(x_{t-1}|x_t)$ can also be considered as a Gaussian distribution, hence we have

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{4}$$

where μ_θ and Σ_θ are the approximated mean and variance, respectively. In this work, we follow Ho *et al.* [8] by simplifying the variance using the fixed β_t^2 as well as applying the reparameterization trick on the mean, then using a neural network model to predict the noise ϵ added to the data rather than optimizing p_θ. However, without properly guiding the diffusion model, the reconstructed data can contain information about the public attribute of any random class, which does not serve the purpose of white-listed data obfuscation. Therefore, we aim to introduce a condition z during the backward diffusion process to guide the diffusion model toward generating a data sample that strongly correlates to the same public attribute of the raw data x_0. We provide more details about how we obtain the condition z in Section 3.2. The model used to predict the noise can be expressed as $\epsilon_\theta(x_t, t, z)$. We implement ϵ_θ based on an open-source UNet architecture released by OpenAI [3]. The timestep t is first embedded using positional encoding techniques proposed by Vaswani *et al.* [22], then the embedded timestep t and condition z are projected through a linear layer into y_t and y_z, respectively. y_t and y_z are conditioned to the UNet using the adaptive group normalization (AdaGN) technique proposed by Dhariwal *et al.* [5]:

$$\mathrm{AdaGN}(h, y) = y_t \mathrm{GroupNorm}(h) + y_z, \tag{5}$$

where h is the output of the first convolutional layer in the UNet's residual block. The predicted noise will be compared with the noise introduced in the forward diffusion process and optimized through a mean squared error (MSE) loss:

$$\mathcal{L}_\theta = \|\epsilon - \epsilon_\theta(x_t, t, z)\|_2^2 = \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}}x_0 + \sqrt{1-\bar{\alpha}}\epsilon, t, z)\|_2^2, \tag{6}$$

where x_0 is the raw sensor data, ϵ is the noise randomly drawn from the Gaussian distribution $\epsilon \sim \mathcal{N}(0, \mathrm{I})$, t is the timestep uniformly sampled between 1 and T.

Once the diffusion model is trained, white-listed obfuscation can be performed by sampling a noisy data point x_T from a Gaussian distribution and denoising it following the backward diffusion process. The original sensor data is fed into the surrogate utility model to extract latent representations of the public attribute to guide the obfuscation process. The obfuscation model outputs a sensor data segment that shares similar public attribute information as the raw sensor data but corresponds to arbitrary private attributes.

3.2 Surrogate Utility Model and Decoder

To achieve white-listed obfuscation, the data generated by the diffusion model must correlate with the public attribute y that can be inferred from the raw sensor data x_0. Without proper guidance, a vanilla diffusion model would blindly sample from the entire training dataset and generate data samples corresponding to possibly different public attribute classes than the user's actual public attribute. Therefore, for each user-desired public attribute, we adopt an auxiliary surrogate utility model $f_\phi(x_0)$, parameterized by ϕ, to extract feature representation z that is specific to the public attribute of x_0, and use it as the condition to guide the generation of obfuscated data. The surrogate utility model is trained to predict the public attribute class y from the raw sensor data x_0. To obtain the public attribute condition z, we run attribute inference using a pretrained model f_ϕ on x_0 and collect the output before the last fully connected classification layer. In our work, we adopt a convolutional neural network-based encoder architecture as the surrogate utility model. We illustrate the workflow of extracting the public attribute condition in Figure 2. Note that the surrogate utility model can have any arbitrary architecture, and the optimal architecture should be determined based on the characteristics of the specific dataset. We also argue that the obfuscated data generated by a simple surrogate utility model can be used by more complicated target models offered by service providers and we demonstrate the results in Section 4.

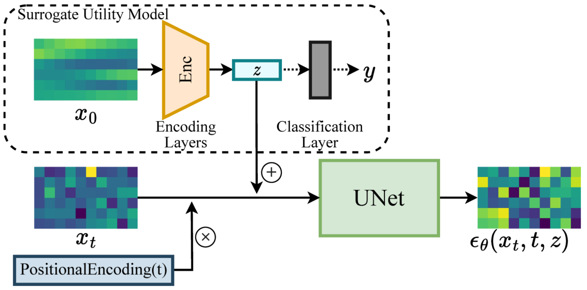


Figure 2: Guiding the diffusion-based obfuscation model.

The surrogate utility model is comprised of two stacked convolutional layers to extract feature maps from the multi-channel sensor readings, with each convolutional layer followed by a ReLU activation function. The two convolutional layers use 16 and 32 kernels of size 2, respectively. The convolutional feature maps are then flattened and fed to three stacked fully connected (FC) layers, each followed by a ReLU activation function. The three stacked FC layer gradually scales down the feature map to a size of 512, 128, and 5, respectively. Lastly, a fully connected classification layer with a softmax activation function maps the latent feature z to the probability distribution of public attribute classes.

We use the cross-entropy loss and Adam optimizer to train the surrogate utility model:

$$\mathcal{L}_\phi = -\sum_{i=1}^{C} y_i \log y_i'. \tag{7}$$

Here C is the number of classes of the white-listed public attribute, y is the true public attribute class, and y' is the predicted public attribute class. This encoding architecture allows the surrogate utility model to selectively compress the rich information embedded in the raw sensor data to enhance the correlation with the white-listed public attributes. Meanwhile, the limited latent feature dimension forces the model to forget information about other attributes that might be considered private. We pretrain the surrogate utility model before training the diffusion model and use the latent output z to guide the diffusion process toward generating data that only contains information about the white-listed public attribute.

4 EVALUATION

4.1 Black-listed Obfuscation Baselines

Baseline 1 - Anonymization Autoencoder (AAE) [15]: AAE is an autoencoder-based obfuscation model. It uses multiple neural network-based regularizers (for public and private attributes) to control the data reconstruction process of the autoencoder. The encoder module of AAE consists of four stacked convolutional layers, each followed by a batch normalization layer. Its decoder module reverses the encoder layers using batch-normalized transposed convolutional layers. We use the code published by the authors on GitHub [2].

Baseline 2 - ObscureNet [7]: ObscureNet is an obfuscation model based on a conditional variational autoencoder (CVAE), which is shown to achieve better privacy-preserving performance than other obfuscation models that are based on autoencoders. The CVAE is jointly trained with an auxiliary classifier that predicts the black-listed private attribute through a minimax game. The private attribute label is used to condition the decoder, enabling the user to provide a random private attribute label in the obfuscation phase to generate synthetic data that fools the auxiliary classifier. Note that a dedicated CVAE must be trained to synthesize data that correspond to each public attribute class. We implement ObscureNet using the code published by the authors on GitHub [1] and use the randomized obfuscation technique as our baseline.

4.2 Dataset

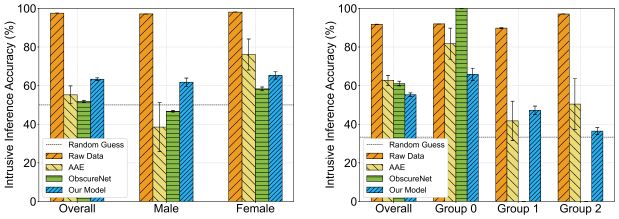
MobiAct is an IMU-based human activity recognition (HAR) dataset that is collected using the 3-axis accelerometer, 3-axis gyroscope, and orientation sensor embedded in a Samsung Galaxy S3 smartphone [4]. From a total of 66 participants that perform 12 activities, we follow our second baseline [7] and select the same 36 users (20 male and 16 female) and 4 activities (walking, standing, jogging, and walking up the stairs) for a fair comparison. We pre-process the data by adopting standardization and segmenting the sensor readings using a sliding window of 128 samples and a stride of 10 samples The dataset is divided using an 8:2 ratio for training and testing. The MobiAct metadata contains three attributes that can be used for evaluating the white-listed and black-listed obfuscation models, namely the user’s activity, gender (binary), and weight group (ternary). Following [7], we categorize a user’s weight into one of the three weight groups, under 70 kg (group 0), between 70 and 90 kg (group 1), and above 90 kg (group 2). The distribution of the 3 weight groups is 18:14:4. We consider activity as the public attribute, and let gender and weight group be the private attributes.

4.3 Evaluation Metrics

We use two *evaluation models*, called intrusive and desired inference models, to evaluate the performance of obfuscation models.

4.3.1 Impact on Privacy. We measure the privacy-preserving ability of an obfuscation model using the private attribute(s) classification accuracy obtained by an intrusive inference model on the obfuscated sensor data. The intrusive inference model is built upon a convolutional neural network (CNN) consisting of 4 convolutional layers followed by 3 fully connected layers. We pretrain the intrusive inference model on raw sensor data, so its accuracy on the obfuscated data is a measure of privacy loss. An ideal obfuscation model should generate obfuscated data that yield the same intrusive inference accuracy as random guessing (50%/33.3% for binary/ternary attributes).

4.3.2 Impact on Data Utility. We define the utility of data by measuring the public attribute classification accuracy of a desired inference model on the obfuscated sensor data. The desired inference model has the same architecture as the intrusive inference model, and is pretrained on the raw sensor data. The sensor data obfuscated by an ideal obfuscation model should achieve nearly the same desired inference accuracy as the raw sensor data.



(a) Gender obfuscation

(b) Weight obfuscation

Figure 3: Intrusive inference accuracy on MobiAct dataset for gender and weight obfuscation. AAE and ObscureNet are trained to protect the gender attribute.

4.4 Privacy Preserving Performance

Given that AAE and ObscureNet are black-listed obfuscation models, we train both models to obscure the gender attribute and study how the obfuscated data could protect the black-listed gender attribute and the unspecified weight attribute.

Figure 3a compares the privacy-preserving performance of the proposed data obfuscation model and the two baselines on gender obfuscation, with results being averaged over 5 runs and the error bar indicating the standard deviation. For reference, we also present the intrusive inference accuracy of the evaluation model on the raw data, as discussed in Section 4.3. Among the three obfuscation models, ObscureNet shows the best gender obfuscation performance with an average intrusive inference accuracy and F1 score of 51.84%, which is pretty close to the level of random guessing. AAE achieves an average gender inference accuracy (F1 score) of 55.23% (54.06%), with higher standard deviation across the 5 runs. This superb obfuscation performance is expected because both ObscureNet and AAE are trained to synthesize data using a dedicated classifier for the user-specified private attribute, i.e. gender. Our proposed white-listed obfuscation model achieves an average gender inference accuracy (F1 score) of 63.33% (63.22%), reducing the gender inference accuracy on raw data by 34.2%. Additionally, our model gives balanced protection for male and female users, achieving 61.78% and 65.28% intrusive inference accuracy, respectively. Although the gender inference accuracy of our white-listed obfuscation model is slightly higher than the two black-listed baselines, it still yields acceptable privacy protection for both genders, without using the knowledge of the private attribute(s) during training.

Next, we study the effectiveness of protecting the weight group attribute. In this case, we keep gender as the private attribute for both AAE and ObscureNet, and study to what extent they can protect another attribute that may be deemed private after the obfuscation models are trained. As Figure 3b shows, the average weight inference accuracy of the respective evaluation model on the data obfuscated by AAE and ObscureNet is greater than 60%. More importantly, we find that all data obfuscated by ObscureNet and most data obfuscated by AAE are classified into the weight Group 0, which is the majority weight group class, representing the weight of 50% of all users in the training set. Although it can be difficult for an adversary to infer the weight of users from weight Groups 1 and 2, the two baselines fail to provide privacy protection for the majority of users. The proposed white-listed obfuscation model, however, shows the best weight obfuscation capability with an average accuracy (F1 score) of 55.30% (49.97%). Similarly, weight

Activity	Raw Data	AAE	ObscureNet	Our Model
Walking	98.10	97.58	95.06	93.90
Standing	99.58	99.58	99.60	99.46
Jogging	99.74	98.26	98.44	98.36
Upstairs	95.14	75.14	92.78	80.34
Overall	98.80	97.71	97.35	96.59

Table 1: Activity recognition accuracy on MobiAct dataset. Gender inference was black-listed when training AAE and ObscureNet. Activity inference was white-listed in all models.

Group 0 suffers from the lowest privacy protection with an average accuracy of 65.8%, yet it still reduces the risk of privacy leakage by nearly 16% and 34% compared to AAE and ObscureNet, respectively. The discrepancy in the weight obfuscation performance can be partly attributed to the non-uniform distribution of weight groups in the training dataset (18:14:4). As discussed in Section 5, we will explore balancing the distribution of private attributes in the training dataset to achieve a more balanced obfuscation performance in future work. To conclude, while the baselines can effectively protect data privacy when users clearly specify the private attribute, they cannot conceal other attributes that the user might consider private in the future. Our proposed white-listed obfuscation model shows competitive obfuscation performance for multiple private attributes, without requiring prior information about them.

4.5 Maintaining Data Utility

We study the utility of the obfuscated data and present the results in Table 1. We keep the black-listed private attribute used to train AAE and ObscureNet the same as in the previous section. We find that the evaluation model for the public attribute (i.e. activity) performs almost the same on the data obfuscated by all three models, with ObscureNet achieving the highest HAR accuracy (F1 score) of 97.35% (85.35%). Inspecting each activity class, ObscureNet outperforms AAE and our proposed model mainly due to a better performance in recognizing the 'walking upstairs' activity. This is due to the fact that ObscureNet requires training a dedicated model for each public attribute, whereas both AAE and our proposed approach train a single model to obfuscate the data regardless of the value of its public attribute. Therefore, AAE and our proposed model might confuse activities with similar movement patterns, such as walking and walking upstairs, in a small number of cases as we observed in the confusion matrix. Nevertheless, the average HAR accuracy (F1 score) of the data obfuscated by AAE and our proposed model is respectively 97.71% (87.91%) and 96.59% (82.43%), suggesting that our model maintains data utility thanks to the guidance provided in the diffusion process.

Overall, our results confirm that the proposed white-listed obfuscation model achieves performance that is on par with black-listed obfuscation models with respect to data utility. Yet, it can effectively obscure attributes that are not white-listed at training time and enjoys more stable training than generative adversarial models.

5 DISCUSSION

Although our experiments demonstrate the feasibility of using the diffusion model to enable white-listed sensor data obfuscation, there remain many open challenges that need to be addressed in future work to pave the way for providing long-term, future-proof privacy protection. We discuss these challenges below:

- The private attribute distribution in the training set is postulated to be sufficiently diverse and reasonably balanced. However, when the private attribute distribution in the training set is imbalanced, the obfuscated data can contain a skewed private attribute distribution and even reveal the distribution in the training set (which is the case in our experiments of disguising weight). Our future work aims to enable a more balanced sampling of the private attribute regardless of the distribution in the training set.
- Our experiments only consider white-listing a single public attribute (i.e., activity). In practice, service providers may need to detect multiple public attributes to enable various services. Our future work will explore white-listing multiple attributes. This can be accomplished by incorporating multiple surrogate utility models and concatenating the conditions, or cascading multiple conditioning processes during the backward diffusion.
- The implicit entanglement of public and private attributes in the sensor data creates non-trivial challenges for preserving user privacy while retaining high data utility. Such challenges are hard to tackle when public and private attributes are strongly correlated. This calls for developing a technique that allows users to navigate the privacy-utility trade-off.
- Recall that we use the output of the last encoding layer that comes before the classification layer in the surrogate utility model to condition the diffusion model. This is because we believe this latent representation is strongly correlated with the public attribute and our experiments suggest that it contains more information about the sensor data compared to using the logits or the public attribute class label, as in classifier-free guidance. Besides, it is more computationally efficient than classifier-guided diffusion since we do not need to compute the gradients of an auxiliary model during the obfuscation process. Although a higher dimensional latent representation provides more information about the public attribute, it may also reveal more information about the private attributes. More experiments are warranted to determine the number of timesteps in the diffusion process and the best learned representation to guide the diffusion model.

6 CONCLUSION

Privacy is an emerging topic in the CPS and IoT community due to the recent advances in pervasive sensing and deep learning. In this work, we explored a white-listing approach to sensor data obfuscation using the guided diffusion process. Specifically, we conditioned a denoising diffusion model using the latent features extracted by a surrogate utility model. We evaluated this obfuscation model on an HAR dataset and compared it with two state-of-the-art obfuscation models that require black-listing the private attributes. We corroborated that by simply white-listing the user-specified public attribute, our proposed model can effectively protect multiple private attributes, without assuming the knowledge of the private attributes.

ACKNOWLEDGMENTS

This research was supported by funding from the Natural Sciences and Engineering Research Council of Canada (RGPIN-2019-04349).

REFERENCES

[1] 2021 [Online]. ObscureNet Implementation. https://github.com/sustainable-computing/ObscureNet. Accessed in 2021.

[2] 2023 [Online]. Anonymization Autoencoder Implementation. https://github.com/mmalekzadeh/motion-sense. Accessed in 2023.

[3] 2023 [Online]. UNet Implementation. https://github.com/openai/guided-diffusion. Accessed in 2023.

[4] Charikleia Chatzaki et al. 2016. Human daily activity and fall recognition using a smartphone's acceleration sensor. In *International Conference on Information and Communication Technologies for Ageing Well and e-Health*. Springer, 100–118.

[5] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 8780–8794.

[6] Omid Hajihassani, Omid Ardakanian, and Hamzeh Khazaei. 2022. Anonymizing Sensor Data on the Edge: A Representation Learning and Transformation Approach. *ACM Transactions on Internet of Things* 3, 1, Article 8 (2022), 26 pages.

[7] Omid Hajihassnai, Omid Ardakanian, and Hamzeh Khazaei. 2021. ObscureNet: Learning Attribute-invariant Latent Representation for Anonymizing Sensor Data. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*. 40–52.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

[9] Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).

[10] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. 2022. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2426–2435.

[11] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. 2021. Variational diffusion models. *Advances in Neural Information Processing Systems* 34 (2021), 21696–21707.

[12] Ang Li et al. 2020. TIPRDC: task-independent privacy-respecting data crowdsourcing framework for deep learning with anonymized intermediate representations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 824–832.

[13] Ang Li et al. 2021. DeepObfuscator: Obfuscating Intermediate Representations with Privacy-Preserving Adversarial Learning on Smartphones. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*. 28–39.

[14] Sicong Liu et al. 2019. Privacy adversarial network: representation learning for mobile data privacy. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 4 (2019), 1–18.

[15] Mohammad Malekzadeh et al. 2019. Mobile sensor data anonymization. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*. 49–58.

[16] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).

[17] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. 2022. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10619–10629.

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[19] Nisarg Raval et al. 2019. Olympus: Sensor Privacy through Utility Aware Obfuscation. *Proc. Priv. Enhancing Technol.* 2019, 1 (2019), 5–25.

[20] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. PMLR, 2256–2265.

[21] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).

[23] Benjamin Weggenmann, Valentin Rublack, Michael Andrejczuk, Justus Mattern, and Florian Kerschbaum. 2022. DP-VAE: Human-readable text anonymization for online reviews with differentially private variational autoencoders. In *Proceedings of the ACM Web Conference 2022*. 721–731.

[24] Xin Yang and Omid Ardakanian. 2022. Blinder: End-to-end Privacy Protection in Sensing Systems via Personalized Federated Learning. *arXiv preprint arXiv:2209.12046* (2022).