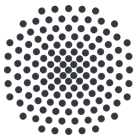


Universität Stuttgart

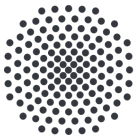
Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Private mouse and keyboard behavioral data

Bachelor Thesis

Author:	Hossam Shehata Shehata Shalaby Elfar
Supervisors:	Prof. Dr. Andreas Bulling
	M.Sc. Mayar Elfares
	M.Sc. Guanhua Zhang
Submission Date:	XX August, 2023



Universität Stuttgart

Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Private mouse and keyboard behavioral data

Bachelor Thesis

Author:	Hossam Shehata Shehata Shalaby Elfar
Supervisors:	Prof. Dr. Andreas Bulling
	M.Sc. Mayar Elfares
	M.Sc. Guanhua Zhang
Submission Date:	XX August, 2023

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowlegement has been made in the text to all other material used

Hossam Shehata Shehata Shalaby Elfar
XX August, 2023

Acknowledgments

First and foremost, I would like to express my deepest and most sincere gratitude to my family for everything they have done for me and for all the love they gave to me. No words can express my love for them.

I would like also to thank my supervisors Prof. Dr Andreas Bulling, Mayar Elfares, M.Sc. and Guanhua Zhang, M.Sc for their support and guidance throughout my bachelor project.
I would like also to thank the German University in Cairo for giving me the opportunity to do my bachelor project at the Institute for Visualization and Interactive Systems, University of Stuttgart.

Abstract

Mouse and keyboard data can be utilized for active authentication [31], personality recognition [37], affective state prediction [9], and predicting user intents [8], making them effective behavioural biometrics in human-computer interaction. Additionally, these data are privacy-sensitive and encompass confidential information about the users in the dataset. This includes sensitive details such as passwords and login credentials, personal messages and communications, as well as banking information, which is legally protected by various privacy laws, including the General Data Protection Regulation (GDPR) [32]. For this purpose, protecting users' privacy has become a crucial problem.

In this thesis, we have developed (1) a remote data science technique as illustrated in Figure 1.1. This technique enables a connection between data owners and data scientists, allowing the latter to access and work with the dataset remotely while respecting the privacy limits defined by the data owner. Importantly, this approach does not involve the data scientist obtaining a copy of the actual data itself. Secondly, We have incorporated (2) the use of the differential privacy technique to create a remote learning approach within Machine Learning. This approach represents an advanced strategy for data querying. The primary goal was to safeguard the individual records in the dataset from identification while enabling analysis and learning from the collective population. By employing differential privacy, we aimed to protect the privacy of the dataset's individuals while still gaining valuable insights by conducting analyses and training models on the dataset as a whole. The results obtained from the task recognition model using the mouse and keyboard dataset are as follows: Baseline Model (No Privacy Guarantee): Accuracy of 75.4%, Strong Privacy Guarantee: Accuracy of 65%, Medium Privacy Guarantee: Accuracy of 65.8%, Low Privacy Guarantee: Accuracy of 69.1%.

Contents

Chapter 1

Introduction

1.1 Motivation

Mouse and keyboard data contain sensitive information such as usernames, passwords, banking information, and social security numbers [34]. and these datasets have potential applications in real-life scenarios. In real-world scenarios, keystrokes and mouse movements can be utilized as biometric features, similar to fingerprints and eye prints, for active user authentication [31] [2]. Many machine learning (ML) applications have also been created using these datasets to model interactive behaviour and enable task recognition and user intent prediction [8] [37] [36], which is helpful for interactive systems and is essential for the creation of adaptable user interfaces (UIs) [11].

However, the data collected is privacy-sensitive, and this behavioral data is vulnerable to attacks [22] that let attackers acquire personal and organizational information and the victim's identity [33]. Hence, techniques to protect individuals' privacy against unwanted inferences are required.

1.2 Problem Statement

The aim of the project is two-fold:

1. To implement a remote data science technique as illustrated in Figure 1.1 that allows data owners to upload their datasets to a domain node that acts like a server while preserving data privacy using differential privacy (DP) and allowing remote processing of mouse and keyboard behavioral data by performing remote data analysis on the network node where data scientists can log into the network and get the required data to run their machine learning models by a limited amount accepted by the data owner [25] [19].

2. Incorporating a Differential Privacy (DP) technique into machine learning (ML) which introduces a more sophisticated strategy for querying and analyzing data. In this approach, each epoch serves as a new query, and effective noise management is crucial due to the accumulated noise throughout the epochs. To address this, we utilized DP-SGD, through which we established a constraint on the information that each epoch can hold about the data. This constraint is set within specific bounds, effectively enabling controlled management of the overall noise of the training process. By employing DP-SGD, we can utilize the machine and deep learning models to extract valuable insights from our behavioral keyboard and mouse dataset while guaranteeing the safeguarding of individual users' privacy and maintaining dependable accuracy [1].

1.3 Contribution

Our contributions encompassed several pivotal aspects, including:

- Establishing a secure connection framework between data scientists and data owners via PySyft.
- Developing a task recognition model based on mouse and keyboard data.
- Conducting a comprehensive comparison of model results for privacy-preserving and non-privacy settings.

Figure 1.1: Remote data science

Chapter 2

Literature Review

2.1 Introduction

Interactive behaviour, such as mouse and keyboard data, plays a crucial role in the evolving field of human-computer interaction (HCI) [36]. These behavioral data help researchers gain insights into how individuals behave and interact with computers [35], enabling advancements in user interface design, usability, and overall user experience. However, the privacy of these datasets is of utmost importance. Given the sensitive nature of behavioral data, protecting individuals' privacy becomes a critical consideration. Without adequate privacy measures, there is a risk of exposing personal information, potentially leading to privacy breaches, identity theft, or unauthorized profiling [15].

Currently, there is limited focus on addressing the privacy concerns specific to mouse and keyboard data. Traditional methods such as anonymization, data swapping, and perturbation as will be discussed in Section 2.3, which are commonly used to protect user information, may not be sufficient in the context of behavioral datasets like mouse and keyboard data. These standard techniques may not effectively preserve privacy while maintaining the utility and accuracy of the data.

Recently, there have been developments in application-specific techniques designed to preserve the privacy of collecting and transmitting mouse and keyboard data [30], particularly in the context of active authentication [12] applications. However, this technique is data-specific, and there is a need to develop a more general approach to protect the privacy of datasets that can be utilized across various contexts.

2.2 Remote data analysis

Privacy concerns in the context of remote data analysis have a longstanding history and encompass a wide range of issues [6, 20]. As the collection and curation of data grow increasingly, potent and electronic data regarding individuals becomes more intricate, and

the need for a robust, substantiated, and mathematically sound concept of privacy becomes imperative. Additionally in the context of remote data analysis, specific challenges and hazards to user privacy emerge, raising several important privacy issues:

- Data Leakage occurs when remote data analysis involves transmitting and processing data outside the secure boundaries of the data owner's infrastructure. The potential for data leakage or unauthorized access increases, posing risks to the confidentiality and integrity of sensitive information [28].
- Remote data analysis requires sharing data with external entities. This raises concerns about who has ownership and control over the data, as well as how it is handled, stored, and potentially shared with third parties.
- Remote data analysis allows inferring sensitive information about individuals even without directly accessing the raw data. Aggregated or derived results may unintentionally reveal private details, leading to privacy breaches [27].
- Remote data analysis often relies on data anonymization or de-identification techniques to protect privacy. However, the risk of re-identification or the possibility of combining data with external sources to infer identities remains a concern as we'll discuss in Section 2.3 [21].

Addressing these privacy concerns requires robust security measures and privacy-preserving techniques. Currently, there are ongoing efforts and research focused on utilizing remote data science [38] to ensure privacy. For instance, the Canada-USA trade dataset [13] has been addressed using this technique. To preserve the privacy of the dataset, it was shared through a domain node, allowing data scientists to access and analyze it remotely without directly obtaining a copy of the data. This approach ensures that sensitive information is protected while enabling valuable insights and analysis to be derived from the dataset. Additionally, the application of differential privacy to training neural networks that utilize behavioral data has been explored in the work by Abadi et al. [1]. In their research, they introduced the dp-sad algorithm, which incorporates differential privacy to enhance the privacy guarantees during the training process. The algorithm was applied to popular datasets such as MNIST and CIFAR, demonstrating the feasibility of training neural networks while preserving the privacy of the underlying behavioral data.

2.3 Privacy-preserving techniques for datasets

In this section, we will delve into various techniques associated with preserving the privacy of datasets. We will explore the risks and concerns associated with each technique, highlighting why solely relying on some of these techniques may not provide sufficient security to safeguard users' information privacy and why using differential privacy offers a robust approach.

2.3.1 Anonymization

To prevent people from being identified, anonymization techniques try to delete or modify personally identifying information (PII) in datasets. Techniques like generalization, suppression, or randomization of data properties can fall under this category [17]. However, "Data Cannot be Fully Anonymized and Remain Useful" [16]. Generally speaking, the richer the data, the more interesting and useful it is. Additionally, there is a risk of re-identification, where external data sources or advanced algorithms can potentially link anonymized data back to individuals, compromising their privacy:

NYC Taxicab Dataset and side-information attacks:
The NYC Taxicab Dataset is a collection of taxi trip records from New York City, containing information about trips made by yellow and green taxis. This dataset includes details such as pickup and drop-off locations, timestamps, trip distances, fare amounts, and more.
Side-information attacks, in the context of datasets like the NYC Taxicab Dataset, generally involve using auxiliary information to infer sensitive or private information about individuals or entities present in the dataset. This auxiliary information could come from various sources, such as social media, public records, or other data leaks. By combining the auxiliary information with the information present in the dataset, attackers can potentially de-anonymize or re-identify individuals in the dataset and extract sensitive information. Vijay Pandurangan [22] delved further into the dataset's information. He managed to exploit details such as license numbers and driver identifiers provided in the dataset, leading to the violation of sensitive information.

Netflix Prize and Linkage attacks
Between 2006 and 2009, Netflix hosted a contest, challenging researchers to improve their recommendation engine. Netflix provided a training dataset of user data to assist teams in designing their strategies. Each data point in the dataset included an anonymized user ID, movie ID, rating, and date. Netflix assured its users that the data had been appropriately de-anonymized to protect individual privacy. Unfortunately, Narayanan and Shmatikov [19] showed that this simplistic method of anonymization was insufficient to protect user privacy, and they managed to match user data from the Netflix dataset with IMDb which was de-identified as illustrated in Figure 2.1. It turns out that this approach was sufficient to re-identify many users from only a few weak matches and this is a type of attack known as linkage attack [18].

2.3.2 Data Swapping

Data swapping includes swapping attribute values between individuals while maintaining the dataset's general statistical characteristics. This makes it very challenging to associate a single person with a set of ideals [14] [24]. However, if the swapping process is reversible or if auxiliary information is available, there remains a risk of re-identification, potentially

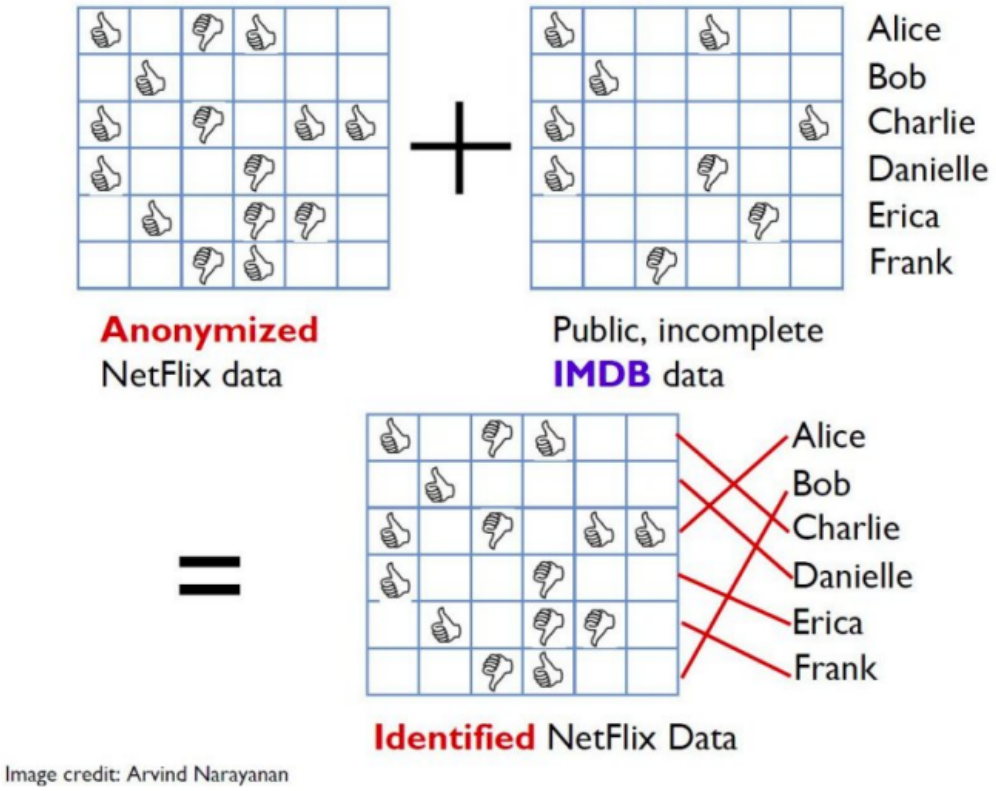


Figure 2.1: Figure illustrating Linkage attack[19]

exposing users' sensitive information [23]. Moreover, in terms of accuracy, swapping data values can disrupt patterns and relationships within the dataset, potentially leading to a loss of valuable analytical insights. This may affect the effectiveness and accuracy of data analysis tasks, limiting the utility of the modified dataset.

2.3.3 Perturbation

To prevent the identification of specific individuals, perturbation introduces controlled noise or randomization to the dataset. This can be accomplished by adding arbitrary values or changing current values while staying within acceptable ranges [3]. The precision and dependability of data analysis results, however, may be negatively impacted by excessive noise or transformations, while inadequate perturbation may not provide sufficient privacy protection. Careful calibration and evaluation of the perturbation parameters are necessary to achieve the desired level of privacy without compromising the accuracy of the data.

2.3.4 Differential privacy

Differential privacy stands as a robust mathematical framework, tailor-made for ensuring privacy guarantees in data analysis. Its primary objective revolves around safeguarding the privacy of individuals whose data is embedded within a dataset. This protective mechanism is harmoniously balanced with the ability to conduct insightful analysis and extract significant statistical information from the dataset.
Two seminal papers, titled "Calibrating Noise to Sensitivity in Private Data Analysis" [7] and "Differential Privacy" [6], emerged in 2006 as the pioneering discussions on this concept, authored by Cynthia Dwork et al. In these notable works, Dwork and her colleagues introduced the revolutionary notion of "differential privacy," a robust mathematical framework that formally articulates and achieves privacy in data analysis. In essence, differential privacy posits that, based on their definition, for any two adjacent datasets differing only by a single data point, the likelihood of an outcome stemming from a specific mechanism M must be bounded. This bound is intricately linked to the privacy loss parameter epsilon (ϵ), where a lesser epsilon ensures heightened privacy assurances.

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(D') \in S]$$

- $\Pr[\mathcal{M}(D) \in S]$: represents the probability of event $M(D)$ happening on dataset D, where M is some function or algorithm, and S is some set of outcomes.
- $\Pr[\mathcal{M}(D') \in S]$: represents the probability of the same event happening on a neighbouring dataset D'.
- ϵ is the privacy parameter, controlling the amount of noise introduced into the computation to ensure privacy.

Mathematically, if we denote the outcome of mechanism M on the dataset, D as $M(D)$ and the outcome on a neighbouring dataset D' as $M(D')$, differential privacy ensures that the probability of $M(D)$ and $M(D')$ deviating significantly is limited and these two outcomes are approximately similar. The level of deviation is controlled by the privacy loss parameter Epsilon (ϵ). When epsilon is set to 0, the outcome of $M(D')$ should be the same as the outcome of D, providing perfect privacy. However, setting epsilon to 0 may result in reduced utility or information loss, as the noise added to preserve privacy can affect the accuracy or fidelity of the results. This can be illustrated as shown in Figure 2.2. As per the definition, the mechanism's outcome should remain relatively unchanged regardless of the inclusion or exclusion of any individual record within the dataset.
In summary, differential privacy is a quantitative mechanism that ensures that the inclusion or exclusion of an individual record in a dataset does not have a substantial impact on the outcome of a given mechanism or computation.

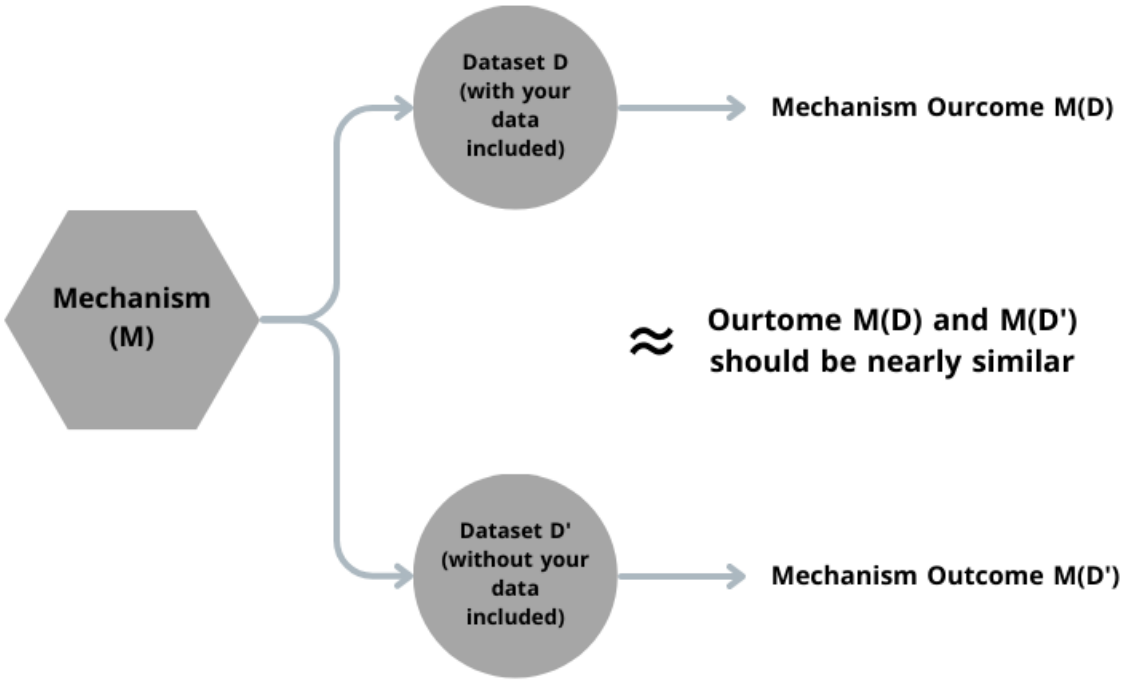


Figure 2.2: Figure illustrating Differential Privacy [6]

Chapter 3

Methods

3.1 Datasets

This section lists mouse and keyboard datasets and explains why we specifically chose the EMAKI dataset to test our method on.

3.1.1 Buffalo Dataset

The "Shared keystroke dataset for continuous authentication" by Sun et al. [29] also referred to as the "Buffalo dataset", offers about 5.5 M mouse actions and 2.5 M keystrokes from 148 participants collected in a laboratory setting. Besides transcription of a predefined text, the participants also performed the task of free text routine work like replying to emails or answering questions.

3.1.2 BEHACOM Dataset

BEHACOM dataset by Sánchez et al [26] aims to provide a set of heterogeneous features that model the users' behaviour when they interact with their personal computers. With that goal in mind, this dataset considers the applications usage statistics, mouse and keyboard actions, and resource consumption of twelve users interacting with their computers for fifty-five days.

3.1.3 EMAKI Dataset

The dataset that was utilized in our method is the "Everyday Mouse And Keyboard Interactions" (EMAKI) dataset by Zhang et al [35] which includes 1.2 M mouse actions and 210 K keystrokes. The data was gathered using a web application and was hosted on a university server. It included three tasks: text entry and editing, image editing,

and questionnaire completion. 52 participants through university mailing lists and social networks. 12 participants who did not finish the study and one teenage participant were filtered out, leading to 39 participants in the end (18 female, 18 male and 3 "other gender"). Their ages ranged between 18 and 54 years (M = 25.05, SD = 6.51). Participants completed the study from 16 countries. On average, they reported having used a mouse and keyboard for 13.64 years (SD = 6.80). 15 participants used laptop touchpads, while the others used traditional mice. 28 participants used laptop keyboards and the rest used standalone keyboards.
The meaning of some of the columns in the dataset is as follows:

1. User: The user column represents the unique identifier assigned to each participant in the study. The values range from 1 to 39, indicating the different users involved.

2. Task: The task column indicates the specific task performed by the user. The tasks are categorized as follows:

   - Skill Test: Tasks numbered 1 and 2, involve skill assessment.
   - Main Tasks: Tasks numbered 3 to 5, are the primary tasks performed by the users.

3. Session: The session column signifies that certain tasks may have multiple sessions. It distinguishes different iterations or divisions of a task performed by a user.

4. The type column represents the type of interaction recorded during the study. Possible values include:

   - "mousemove": Captures mouse movement events.
   - "mousedown": Records when the user presses the mouse button.
   - "mouseup": Records when the user releases the mouse button.
   - "keydown": Indicates the pressing of a keyboard key.
   - "keyup": Indicates the releasing of a keyboard key.

5. Value: The value column provides additional information based on the type of interaction recorded. For mouse events (mousedown, mouseup), it specifies whether the left or right mouse button was clicked. For keyboard events (keydown, keyup), it represents the value of the pressed or released key.

6. Resolution: The resolution column denotes the screen resolution of the user's device during the study. It can be used for normalization purposes, particularly for mouse coordinates.

7. OCEAN: The OCEAN column refers to the Big Five personality traits (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism). The values are divided into high and low based on their median value. This information allows for potential analysis and correlation between personality traits and user behaviour.

[12]:

	ID	user	session	task	INVALID	type	timestamp	X	Y	value	resolutionX	resolutionY	mu	O	C	E	A	N
0	173183	1	1	3	0	mousemove	1616056087883	0.477391	0.748698	NaN	1349	768	1_8	0	1	1	1	1
1	173184	1	1	3	0	mousemove	1616056087900	0.476649	0.744792	NaN	1349	768	1_8	0	1	1	1	1
2	173185	1	1	3	0	mousemove	1616056087917	0.474426	0.739583	NaN	1349	768	1_8	0	1	1	1	1
3	173186	1	1	3	0	mousemove	1616056087933	0.470719	0.733073	NaN	1349	768	1_8	0	1	1	1	1
4	173187	1	1	3	0	mousemove	1616056087949	0.467013	0.721354	NaN	1349	768	1_8	0	1	1	1	1

Figure 3.1: Sample of the dataset [35]

Considering these options, we have chosen the EMAKI dataset as the ideal candidate for testing our technique for several reasons. All these datasets were collected in constrained laboratory settings, except for the EMAKI dataset, which takes a different approach. EMAKI represents a move towards fully unconstrained settings, enabling the inclusion of more natural and interactive behaviors in the dataset [35]. Also, the participants' pool is notably diverse, encompassing individuals from various countries who utilized a range of input devices and screen resolutions. This diversity adds richness to the dataset by capturing a wider spectrum of user behaviors and interactions.

3.2 Pre-processing

In the preprocessing phase, the EMAKI dataset was divided into mouse data and keyboard data to handle them separately. This separation was likely done because the two types of input data, mouse data and keyboard data, may have different characteristics and need specific preprocessing steps.

After dividing the dataset, the next step was to remove any null or missing values from the columns. Null values are data points that are missing or unknown, and they can create issues during data analysis and modelling. By removing these null values, the dataset becomes more consistent and suitable for further analysis.

After removing the null values from the dataset and before proceeding with the analysis, the next step involved encoding the string values into integers. This process is called "label encoding" or "integer encoding," and it's necessary because most machine learning algorithms and statistical models work with numerical data, not with categorical or string data. Label encoding converts categorical variables (strings) into a series of integers. Each unique category is assigned a unique integer value. For example : "keyup" $\rightarrow$ 0, "keydown" $\rightarrow$1, "mousemove" $\rightarrow$ 2, "mouseup" $\rightarrow$ 3, "mousedown" $\rightarrow$ 4. The next step was to group the data by the user. Grouping the data by user is a common practice in data analysis when dealing with user-specific data. Additionally, we need to consider the impact of differential privacy, which ensures that the inclusion or exclusion of any user's data won't significantly change the analysis results or disclose sensitive information about that user. Subsequently, we conducted the standard training test data split and we implemented a sliding window approach to provide the model with more comprehensive information. This allowed the model to recognize tasks based on, for example, 5 seconds of behavior rather than just 0.05 seconds.

3.3 Milestone 1: Remote Data Science with differential privacy

To create a remote data science setup, the PySyft [4] and Hagrid [5] libraries were utilized. PySyft, which is an open-source Python library by OpenMined, offers various tools and utilities for privacy-preserving machine learning techniques like differential privacy. Hagrid, likewise developed by OpenMined, is a federated learning privacy-preserving database. that allows for model training across decentralized and distributed devices or servers while keeping data localized. It provides a privacy-first encrypted data store, allowing secure storage, querying, and handling of sensitive data for data scientists and developers.
As illustrated in Figure 1.1, the remote data science technique setup comprises three main components: The domain server, the data scientist, and the data owner.

Domain server
The Domain Server acts as an intermediary or coordinator in the privacy-preserving setup. Its role is to facilitate the secure exchange of data and computations between the Data Scientist and the Data Owner. The Domain Server implements various privacy protocols, such as Federated Learning or Differential Privacy, to ensure the privacy guarantees of the subjects under study.
After loading our dataset and performing the preprocessing, we proceeded to launch the domain server, which will hold both the data from the mouse and the keyboard. We used the Syft Orchestra to launch a domain server, which will launch the domain server in development mode and allow local access for Data scientists to log into the domain server and access our datasets. The Domain Server can be utilized either through the command line or through the UI shown in Figure 3.2. From the UI, you can view the datasets included in this domain, the users registered to our domain, and their requests to access any piece of data.

Data Owner
A data owner provides datasets that they are willing to make available for research by an outside party they may or may not fully trust has good intentions.

1. The initial step involves the data owner deploying our domain server with a designated name via PySyft, ensuring its availability and readiness for dataset uploading.

2. Load the EMAKI dataset and prepare it for sharing. and ensure the dataset is in a format that PySyft can work with, such as PyTorch tensors.

3. Establish user accounts for data scientists, enabling them to initiate queries and utilize our dataset.

4. Allocate a predefined privacy budget to each data scientist, granting them the ability to query the data within specified limits.

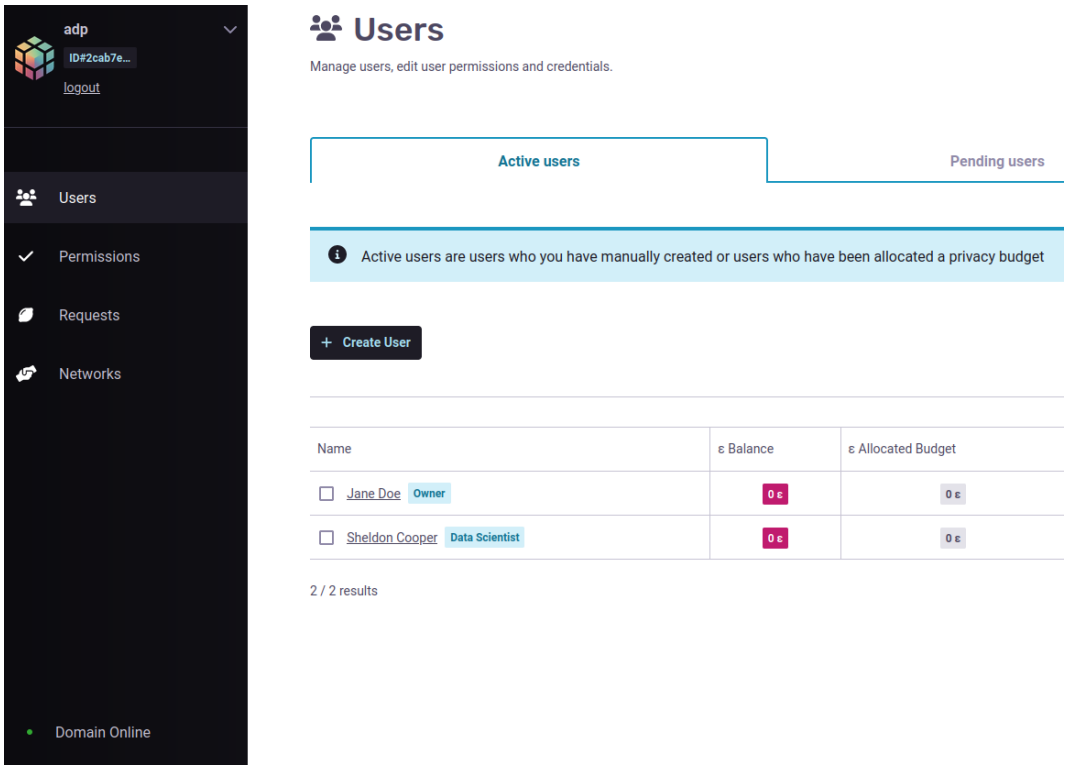


Figure 3.2: Domain Server

5. Review and either approve or decline queries submitted by data scientists.

Data Scientist

Data scientists are the end users who analyze and manipulate datasets and intend to conduct computations or obtain answers to specific questions using datasets owned by one or more data owners.

Once the dataset is uploaded to the domain server, the data owner can leverage their privacy budget to initiate queries on the dataset. However, it's important to note that the data owner's access is restricted to a mock dataset, which is a replicated version of the original data. This mock dataset maintains an identical size and dimensions to the original, ensuring that sensitive information remains confidential during querying processes.

Subsequently, the data owner submits their queries to the domain server for execution on the original dataset. The outcomes are transmitted back to the data scientist, incorporating additional noise through the utilization of Laplacian noise.

$$F(x) = f(x) + Lap\left(\frac{s}{\varepsilon}\right)$$

- s: the sensitivity of the query.
- ε: the privacy loss.
- Lap(x): a sample from the Laplace distribution with scale parameter x.

The sensitivity of a function f is defined as the maximum amount by which the output of f can change when its input changes by 1. For instance, if a data scientist submits a count query to the domain server, the sensitivity of this query is 1. This is because the maximum change in the output of the count query, caused by adding or removing a single individual data record, is 1.
The ϵ is the privacy parameter that controls the amount of noise added to achieve differential privacy.
To safeguard individual privacy while enabling statistical analysis of the data, we leveraged these two hyperparameters to extract a sample from the Laplace distribution. This sample was subsequently added to the output of the function.

Privacy on the domain server
We have upheld privacy on the domain server through various measures:

- Data scientists possess limited access exclusively to the mock dataset ensuring the confidentiality of sensitive data while conducting search operations.
- We have allocated a specific privacy budget to each data scientist for querying the data to mitigate the risk of reconstruction attacks [10].
- We introduce a controlled quantity of Laplace noise to the returned results, which are then transmitted back to the data scientists.

3.4 Milestone 2: Deep Learning with differential privacy

[Deep Learning with differential privacy] In deep learning, we achieve differential privacy with differential private stochastic gradient descent (DP-SGD), by adding noise to the gradients of losses so that each data entry (individual's data) has plausible deniability.

Differential private stochastic gradient descent (DP-SGD)
The DP-SGD algorithm modifies the stochastic gradient descent algorithm, which forms the basis for many standard optimizers in machine learning. Privacy guarantees that can

be mathematically proven are provided through differential privacy for models trained using DP-SGD.
Some primary alterations have been incorporated into the standard SGD to minimize the empirical loss attributed to individual data points within the dataset. These modifications as illustrated in Figure 3.3 encompass :

1. Splitting training data batch into smaller subsets known as microbatches.
2. Clip the gradients of losses computed at each step.
3. Add Noise to the clipped gradients.

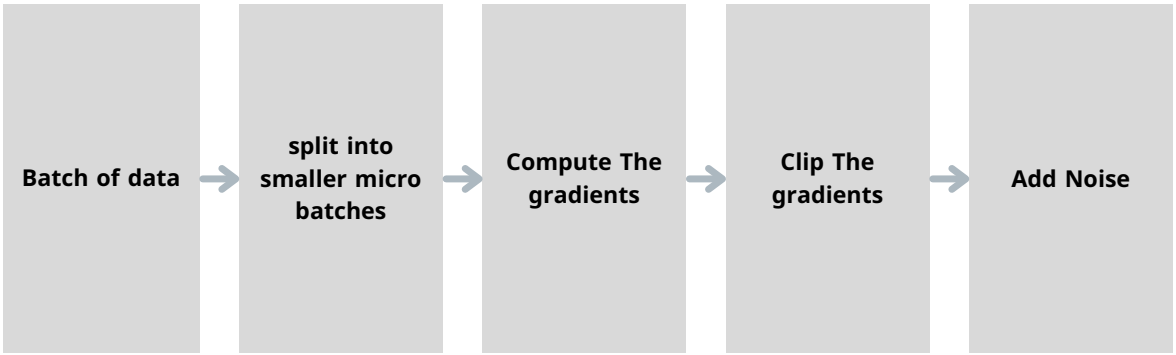


Figure 3.3: DP-SGD setup

Split training data:
In this step, the DP-SGD splits the data from each batch into smaller subsets, which are known as microbatches. These microbatches are further processed to compute gradients, which are used to update the model's parameters. The concept of microbatches is commonly used in distributed training settings, where large datasets are split across multiple nodes or devices to enable efficient training of models, Increasing the number of microbatches will typically enhance your utility while delaying your training in terms of wall-clock time.

Clipping the gradients
In this step, As illustrated in Figure 3.4 the gradients of losses that were computed from

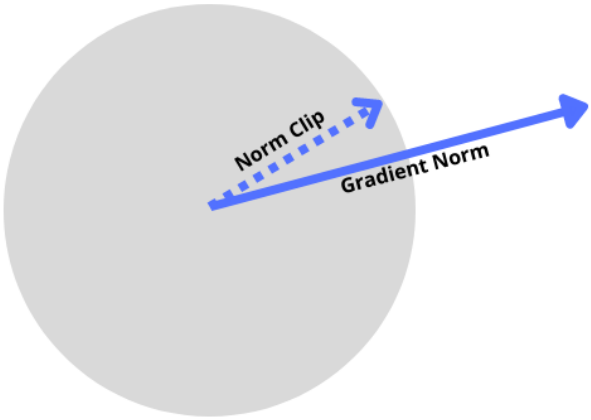


Figure 3.4: Gradient clipping

each microbatch are clipped to a certain threshold, which is known as the clipping norm. The idea behind gradient clipping is to limit the magnitude of the gradients calculated during training. This helps mitigate the potential privacy risks associated with having extreme gradients that could reveal sensitive information about individual data points. Clipping the gradients helps ensure that no single data point has an overly influential impact on the updates to the model's parameters.

Adding Noise
The final step in DP-SGD is to add an amount of noise to the clipped gradients of each micro batch, this noise is a sample from the Gaussian distribution. Adding noise helps in achieving differential privacy by making it harder for an attacker to distinguish the impact of any specific data point on the gradients. This ensures that individual data points don't have a disproportionate impact on the learned model.

Before delving into the integration of DP-SGD and privacy considerations, we initially constructed a basic neural network. The purpose was to contrast the training accuracy of the dataset under two scenarios: one with privacy measures incorporated and one without. We will commence by fine-tuning the hyperparameters, including the clipping norm, the noise magnitude, and the number of microbatches. This iterative process involves evaluating the model's outputs to discern the optimal equilibrium between privacy preservation and accuracy. Taking into consideration two things (1) the computation cost as the process of DP-SGD is an iterative process and going through these steps Figure 3.3 with a high number of microbatches and a small threshold for gradient will decrease the updates. which will generally give us a higher privacy guarantee, but in the end, it will affect the accuracy of our training model. (2) The dataset size, a larger dataset might also require more noise to be added to maintain a consistent level of privacy, potentially affecting the utility of the model.

Chapter 4

Results

4.1 Dataset

We conducted experiments on the EMAKI dataset, as described in the methodology section, for task recognition. The data was split into training and testing sets using an 80% train-test split for each user group. The dataset consisted of 768,256 training examples and 192,000 testing examples. Each example consisted of two attributes: mouse 'x', and 'y' values for mouse data, the "value" attribute for keyboard data, and labels for tasking of 6 tasks. For the experiments, we utilized a simple feedforward neural network with Bidirectional Gated Recurrent Unit (GRU), ReLU units and a softmax activation function with 6 output units corresponding to the 6 possible task classes. The Adam optimizer was employed to update the model's parameters during the training process.

4.2 Experiments

In this section, we showcase the outcomes of our evaluation of the "DP-SGD" (differential privacy-stochastic gradient descent) technique on the EMAKI dataset, encompassing both mouse and keyboard data. Additionally, we conduct a comparison of the overall privacy loss (ϵ, δ) between the neural network without privacy and the neural network after implementing the DP-SGD technique.

In our experiment, we conducted a comparison of the loss and accuracy for three levels of privacy: (1) Large Noise, (2) Medium Noise, and (3) Small Noise

4.2.1 Baseline model

The baseline model consists of three sequential layers. The first layer is a BidirectionalGRU layer with 128 nodes. The second layer is a Dense layer with 64 nodes, and it

utilizes the ReLU activation function for introducing non-linearity. The final layer is another Dense layer with 6 nodes and a softmax activation function, enabling the model to perform multi-class classification with 6 possible classes. We can reach an accuracy of 75.4% in about 100 epochs. With a sliding window of window size (W = 60) and step size (S = 30).

- Learning rate: 10^{-3}.
- Batch size: 256 batches.

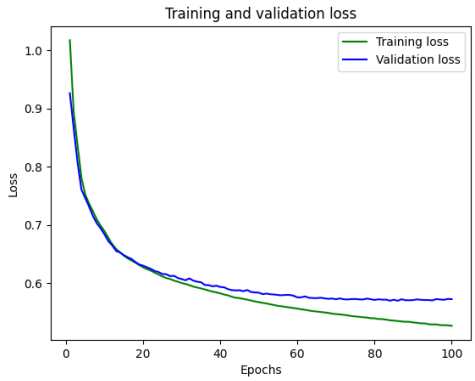


Figure 4.1: Training and validation loss

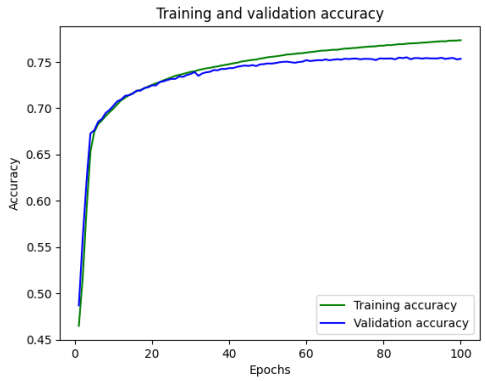


Figure 4.2: Training and validation accuracy

4.2.2 Appling the DP-SGD

Tensorflow optimizer was used to apply the DP-SGD to our dataset, We conducted experiments with the same architecture as the baseline model for the differentially private version. , However, in the differentially private model, we incorporated three hyperparameters to restrict the sensitivity of the training process: num_microbatches, L2_norm_clip, and noise_multiplier. We present the outcomes for three noise scale options, (1) Large noise ($\epsilon = 0.57$, $\delta = 2.4e^{-7}$), (2) Medium noise ($\epsilon = 1.79$, $\delta = 2.4e^{-7}$), and Small noise ($\epsilon = 4.48$, $\delta = 2.4e^{-7}$), here epsilon (ϵ) represents the privacy loss where smaller epsilon guarantees tighter privacy, and delta (δ) represents the failure probability. In all of the three options we set the learning rate to 10^{-3} and the batch size to 256.

Figures [4.3] and [4.4] display the results of the loss function and accuracy for the Large noise ($\epsilon = 0.57$, $\delta = 2.4e^{-7}$)-differential privacy. The model achieved an accuracy of 64% to 65.4% while maintaining tight privacy guarantees. We set the clipping norm to 1.1 and the noise multiplier to 5.8, with 128 micro-batching.

Figures [4.5] and [4.6] display the results of the loss function and accuracy for the Medium noise ($\epsilon = 1.79$, $\delta = 2.4e^{-7}$)-differential privacy. The model achieved an accuracy of 65.2%

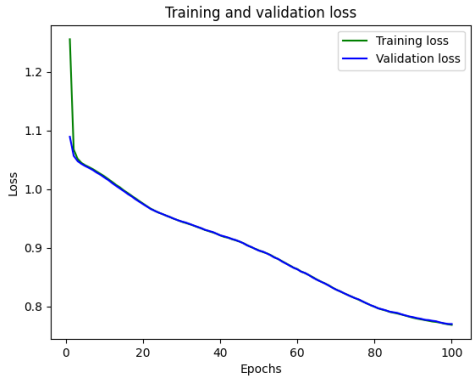


Figure 4.3: Large noise (Training and validation loss)

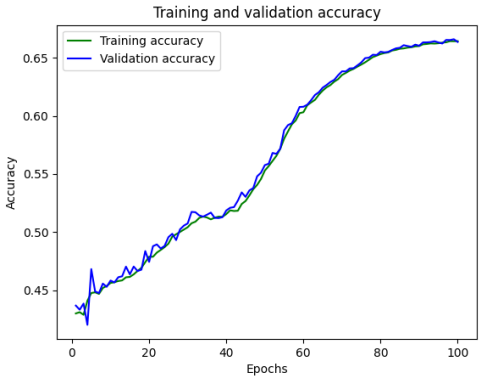


Figure 4.4: Large noise (Training and validation accuracy)

to 66.3% while maintaining good privacy guarantees. We set the clipping norm to 1.5 and the noise multiplier to 2.1, with 128 micro-batching.

Figures [4.7] and [4.8] display the results of the loss function and accuracy for the Small

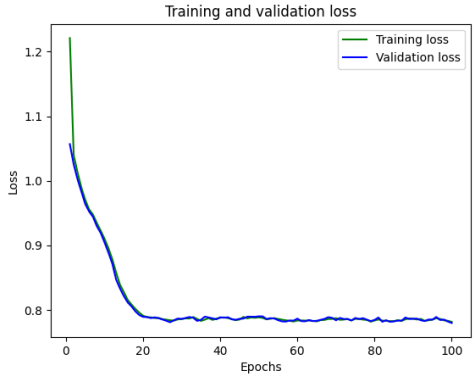


Figure 4.5: Medium noise (Training and validation loss)

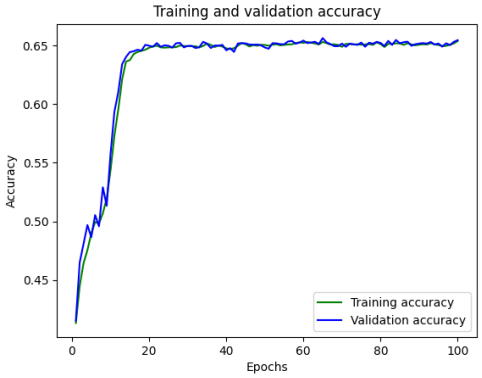


Figure 4.6: Medium noise (Training and validation accuracy)

noise ($\epsilon = 1.79$, $\delta = 2.4e^{-7}$)-differential privacy. The model achieved an accuracy of 69.0% to 71.0%. We set the clipping norm to 1.5 and the noise multiplier to 1.1, with 128 micro-batching.

Utilizing the outcomes from our three options, it has been determined that adjusting the L2_norm_clip with a threshold ranging from 0.5 to 1.5 yields a satisfactory balance between accuracy and privacy. In addition, by analyzing the comparison between the amount of noise introduced to the dataset and the corresponding ϵ value, it is evident that the range of 0.8 to 4.5 as shown in fig. 4.9 provides a significant level of privacy assurance.

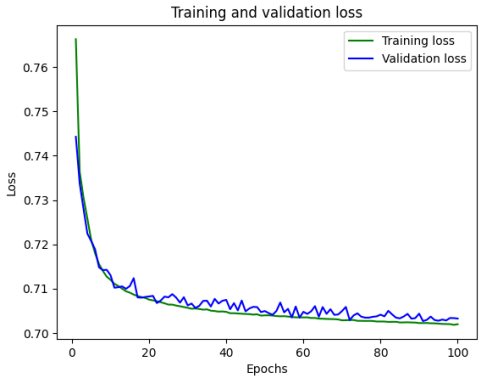


Figure 4.7: Small noise (Training and validation loss)

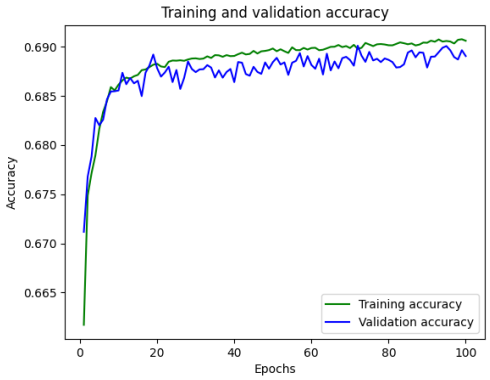


Figure 4.8: Small noise (Training and validation accuracy)

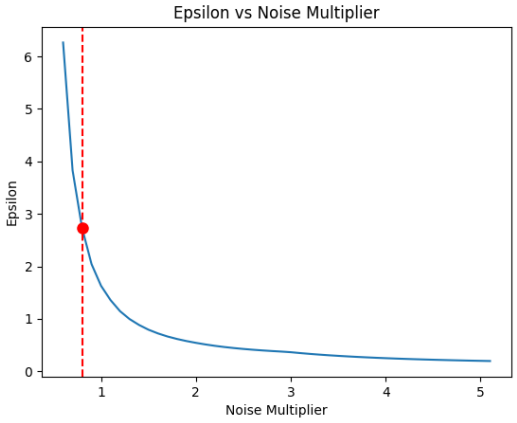


Figure 4.9: Epsilon vs Noise Multiplier

4.3 Discussion

From what we found in our experiments, DP-SGD as outlined in Section 3.4 combines the principles of stochastic gradient descent (SGD), with differential privacy (DP), The key parameter in DP-SGD is epsilon (ϵ), which controls the trade-off between privacy and utility. A smaller epsilon provides higher privacy but might lead to reduced accuracy or utility in the model. This approach applies to both modalities of mouse and keyboard, accommodating different settings and distinct hyperparameters tuning for each modality. Generally, enhancing privacy measures often involves introducing noise, obfuscation, or other mechanisms that can impact the performance of the models being trained. The influence on utility can become more noticeable as privacy measures strengthen. After conducting experiments and fine-tuning the model settings and privacy hyperparameters—such as Norm Clip, Noise Multiplier, and microbatches—we arrived at reasonable levels of privacy. These levels have been categorized into three primary tiers High noise, Medium noise and Low Noise, as depicted in Table 4.1. In the 'High noise' level, there

Table 4.1: Comparison of The Three Levels of Privacy

	Features							
	Batch Size	microbatches	Epochs	Norm Clip	Noise Multiplier	Epsilon	Delta	Accuracy
No Privacy	128	64	100	-	-	-	-	75.4%
High Noise	128	64	100	1.1	5.8	0.57	$2.4e^{-7}$	65%
Medium Noise	128	64	100	1.2	2.1	1.75	$2.4e^{-7}$	65.8%
Low Noise	128	64	100	1.5	1.1	4.48	$2.4e^{-7}$	69.1%

exists a robust privacy guarantee with $\epsilon = 0.57$, ensuring that the impact of any single individual's data on the outcome of the algorithm is limited, Simultaneously, we achieved satisfactory model accuracy, with an average of around 65%. Within the 'Medium noise' level, a reasonable privacy assurance is present with $\epsilon = 1.57$, ensuring that the influence of an individual's data on the algorithm's output is constrained, however with some degree of error, Also, we achieved better model accuracy, with an average of around 65.8%. In the 'Low Noise' level, we attained the highest model accuracy, averaging around 69.1%, which is notably close to the accuracy of the model without any privacy guarantee. This outcome is anticipated since the addition of minimal noise to each clipped gradient implies an $\epsilon = 4.48$. Although this does bring a reduction in privacy, the privacy level remains acceptable.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we focused on addressing the issue of privacy concerning mouse and keyboard data. These behavioral datasets encompass sensitive information about individuals, necessitating the implementation of protective measures to safeguard their data from potential attacks. Our main focus was on two important factors.

Firstly, we created a remote data science setup that makes it simple for these datasets to be published online. This setup facilitates communication between data owners who are eager to share useful behavioral datasets without being constrained by privacy concerns. This platform also makes it possible for other data scientists to access this data for their own research and data science projects. By simplifying the procedure of getting access to the datasets. This established setup has demonstrated its efficacy in terms of usability and scalability. Its usability lies in the fact that data scientists can readily utilize our dataset without requiring an in-depth understanding of differential privacy concepts. Simultaneously, the setup's scalability is evident in its ability to accommodate multiple datasets alongside numerous data scientists utilizing them within our remote data science environment.

Secondly, we developed the remote learning technique, which connects the use of remote data science with Machine learning and allows for a more advanced strategy for querying data. where each epoch is treated as a query, and careful management of accumulated noise is essential for successful and accurate model training. This technique is specifically designed to enhance the privacy of deep learning models utilizing mouse and keyboard behavioral data and shows the balance between privacy and utility when introducing privacy considerations into the training process of deep learning models. Our primary focus was on integrating differential privacy into deep learning models and comparing the model's results with and without privacy. We achieved this by employing

the differential private stochastic gradient descent technique (DP-SGD). This approach enables each data point to contribute to the model's training process to a certain amount while introducing noise. This carefully-calibrated noise serves as a protective barrier; even if an adversary gains access to the model's updates, they are unable to deduce substantial information about individual data points. which allows for the training of models while minimizing the risk of exposing sensitive information in the training data.

5.2 Future Work

Future work can explore the effectiveness of our approach on other modalities like gaze and eye tracking data which can offer valuable insights and contribute to a deeper understanding of human behavior and interaction. Another promising avenue for future work involves applying our approach to real-world scenarios and domains through the development of a practical tool for Differential Privacy Stochastic Gradient Descent (DP-SGD) based on our approach. This tool would leverage the insights and methodologies established in our study to create a user experience for data scientists.

Appendix

List of Figures

List of Tables

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Francesco Bergadano, Daniele Gunetti, and Claudia Picardi. User authentication through keystroke dynamics. *ACM Transactions on Information and System Security (TISSEC)*, 5(4):367–397, 2002.

[3] Keke Chen and Ling Liu. Geometric data perturbation for privacy preserving outsourced data mining. *Knowledge and information systems*, 29:657–695, 2011.

[4] OpenMined Community. Pysyft: A python library for privacy-preserving machine learning. https://github.com/OpenMined/PySyft, 2017.

[5] OpenMined Community. Hagrid: A privacy-preserving database for federated learning. https://github.com/OpenMined/Hagrid, 2021.

[6] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

[7] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[8] Anis Elbahi, Mohamed Ali Mahjoub, and Mohamed Nazih Omri. Hidden markov model for inferring user task using mouse movement. In *Fourth International Conference on Information and Communication Technology and Accessibility (ICTA)*, pages 1–7. IEEE, 2013.

[9] Clayton Epp, Michael Lippold, and Regan L Mandryk. Identifying emotional states using keystroke dynamics. In *Proceedings of the sigchi conference on human factors in computing systems*, pages 715–724, 2011.

[10] Simson Garfinkel, John M Abowd, and Christian Martindale. Understanding database reconstruction attacks on public data. *Communications of the ACM*, 62(3):46–53, 2019.

[11] Jeremy Goecks and Jude Shavlik. Automatically labeling web pages based on normal user actions. In *Procedings of the IJCAI Workshop on Machine Learning for Information Filtering*, 1999.

[12] Richard P Guidorizzi. Security: active authentication. *IT professional*, 15(4):4–7, 2013.

[13] Adam James Hall, Madhava Jay, Tudor Cebere, Bogdan Cebere, Koen Lennart van der Veen, George Muraru, Tongye Xu, Patrick Cason, William Abramson, Ayoub Benaissa, et al. Syft 0.5: A platform for universally deployable structured transparency. *arXiv preprint arXiv:2104.12385*, 2021.

[14] ASM Touhidul Hasan, Qingshan Jiang, Jun Luo, Chengming Li, and Lifei Chen. An effective value swapping method for privacy preserving data publishing. *Security and Communication Networks*, 9(16):3219–3228, 2016.

[15] Athina Ioannou, Iis Tussyadiah, and Yang Lu. Privacy concerns and disclosure of biometric and behavioral data for travel. *International Journal of Information Management*, 54:102122, 2020.

[16] Tiancheng Li and Ninghui Li. On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–526, 2009.

[17] Abdul Majeed and Sungchang Lee. Anonymization techniques for privacy preserving data publishing: A comprehensive survey. *IEEE access*, 9:8512–8545, 2020.

[18] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

[19] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.

[20] Arvind Narayanan and Vitaly Shmatikov. Myths and fallacies of" personally identifiable information". *Communications of the ACM*, 53(6):24–26, 2010.

[21] Gregory S Nelson. Practical implications of sharing data: a primer on data privacy, anonymization, and de-identification. In *SAS global forum proceedings*, pages 1–23, 2015.

[22] Vijay Pandurangan. On taxis and rainbows. Medium, June 27 2014.

[23] Jerome P Reiter. Estimating risks of identification disclosure in microdata. *Journal of the American Statistical Association*, 100(472):1103–1112, 2005.

[24] Mercedes Rodriguez-Garcia, Montserrat Batet, and David Sánchez. Utility-preserving privacy protection of nominal data sets via semantic rank swapping. *Information Fusion*, 45:282–295, 2019.

[25] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017*, 2018.

[26] Pedro M Sánchez Sánchez, José M Jorquera Valero, Mattia Zago, Alberto Huertas Celdrán, Lorenzo Fernández Maimó, Eduardo López Bernal, Sergio López Bernal, Javier Martínez Valverde, Pantaleone Nespoli, Javier Pastor Galindo, et al. Behacom-a dataset modelling users' behaviour in computers. *Data in Brief*, 31:105767, 2020.

[27] Asaf Shabtai, Yuval Elovici, and Lior Rokach. *A survey of data leakage detection and prevention solutions*. Springer Science & Business Media, 2012.

[28] Subashini Subashini and Veeraruna Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, 34(1):1–11, 2011.

[29] Yan Sun, Hayreddin Ceker, and Shambhu Upadhyaya. Shared keystroke dataset for continuous authentication. In *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2016.

[30] Yan Sun and Shambhu Upadhyaya. Secure and privacy preserving data processing support for active authentication. *Information Systems Frontiers*, 17:1007–1015, 2015.

[31] Issa Traore, Isaac Woungang, Mohammad S Obaidat, Youssef Nakkabi, and Iris Lai. Combining mouse and keystroke dynamics biometrics for risk-based authentication in web environments. In *2012 fourth international conference on digital home*, pages 138–145. IEEE, 2012.

[32] W Gregory Voss. European union data privacy law reform: General data protection regulation, privacy shield, and the right to delisting. *The Business Lawyer*, 72(1):221–234, 2016.

[33] Jincheng Wang, Zhuohua Li, John CS Lui, and Mingshen Sun. Topology-theoretic approach to address attribute linkage attacks in differential privacy. *Computers & Security*, 113:102552, 2022.

[34] Dema Zaidan, Asma Salem, Andraws Swidan, and Ramzi Saifan. Factors affecting keystroke dynamics for verification data collecting and analysis. In *2017 8th International Conference on Information Technology (ICIT)*, pages 392–398. IEEE, 2017.

[35] Guanhua Zhang, Matteo Bortoletto, Zhiming Hu, Lei Shi, Mihai Bâce, and Andreas Bulling. Exploring natural language processing methods for interactive behaviour modelling. *arXiv preprint arXiv:2303.16039*, 2023.

[36] Guanhua Zhang, Susanne Hindennach, Jan Leusmann, Felix Bühler, Benedict Steuerlein, Sven Mayer, Mihai Bâce, and Andreas Bulling. Predicting next actions and latent intents during text formatting. In *Proceedings of the CHI Workshop Computational Approaches for Understanding, Generating, and Adapting User Interfaces (2022-01-01)*, pages 1–6, 2022.

[37] Yinghui Zhao, Danmin Miao, and Zhongmin Cai. Reading personality preferences from motion patterns in computer mouse operations. *IEEE Transactions on Affective Computing*, 13(3):1619–1636, 2020.

[38] Alexander Ziller, Andrew Trask, Antonio Lopardo, Benjamin Szymkow, Bobby Wagner, Emma Bluemke, Jean-Mickael Nounahon, Jonathan Passerat-Palmbach, Kritika Prakash, Nick Rose, et al. Pysyft: A library for easy federated learning. *Federated Learning Systems: Towards Next-Generation AI*, pages 111–139, 2021.