**NeurIPS Hide-and-seek Privacy Challenge documentation questionnaire**

**Team name**

|  |
| --- |
| AreYouHiding |

**Submission filenames(s)**

|  |  |
| --- | --- |
| Hider | AreYouHiding\_hider\_new\_ES\_modified |
| Seeker | None |

**What class of algorithms does your solution belong to?** (e.g. GANs, VAEs, noise-injection, nearest neighbor, etc.)

|  |  |
| --- | --- |
| Hider | Undercomplete Autoencoder |
| Seeker | None |

**Describe your algorithm in one sentence** (e.g. “Noise is added to the original data and then this data is returned.”)

|  |  |
| --- | --- |
| Hider | An undercomplete autoencoder that encodes feature representations and reconstructs them |
| Seeker | None |

**Describe your algorithm in words** (e.g. “Noise is drawn from a Gaussian distribution, with mean 0 and variance s, where the dimension is determined by the size of the dataset. This noise is added to the original data to produce a noisy version of the dataset and this noisy dataset is then returned as the synthetic data.”)

|  |  |
| --- | --- |
| Hider | 1. We first preprocessed the data (utils/data\_preprocess.py)  - Impute negative, zeros, and nan values with first quantile (L61-77)  - Pad or truncate sequences to max\_seq\_len (L131-145)  - MinMax Scaling (L178-228)  2. We then train the model (examples/hider/timegan/timegan.py)  - extract\_time (L332)  - MinMaxScale (L335)  - Train the embedding and recovery networks (L368-382)  - No differential privacy was used during the training procedure.  3. We then generate the synthetic data (examples/hider/timegan/timegan.py)  - Return generated data (recovered data) from `ori\_data` using ae\_generate function. (L476, L198-214) |
| Seeker | None |

**Specify any loss functions used** (e.g. “No loss functions used.”)

|  |  |
| --- | --- |
| Hider | Reconstruction loss (MSE) |
| Seeker | None |

**Specify any hyperparameters and how they are optimized (or preset values)** (e.g. “The noise size, s, is set to 0.1.”)

|  |  |
| --- | --- |
| Hider | # Data hyperparameters  max\_seq\_len: 100,  train\_rate: 0.5,  # Model hyperparameters  rnn\_module: gru  num\_layers: 3,  batch\_size: 256  feature\_dim: 71,  z\_dim: 71  hidden\_dim: 10,  training\_iterations: 2000  # Optimizer hyperparameters  optimizer: adam  learning\_rate: 0.001,  epsilon: 1e-08,  warmup\_steps: 2  # Other hyperparameters  random\_seed: 0 |
| Seeker | None |

**Specify any pre-trained models used by your algorithm** (e.g. “None.”)

|  |  |
| --- | --- |
| Hider | None |
| Seeker | None |

**Pseudo-code for your algorithm**

e.g. **Inputs:** Dataset, D, random seed

**Hyperparameters:** s (default 0.1)

1. Determine dataset dimension: n x d x T

2. Draw N ~ N(0, s), an n x d x T dimensional Gaussian

3. Return D + N

|  |  |
| --- | --- |
| Hider | TimeAE:  while train:  1. Sample batch X  2. H = embedder(X)  3. X\_hat = recovery(H)  4. MSE(X, X\_hat) # Recovery Loss  while generate:  1. H = embedder(X)  2. X\_generated = recovery(H) |
| Seeker | None |

Finally, alongside this document **please also submit a commented version of your code**. Please include:

- Docstrings for each new class/function defined

- Inline comments for your main function/class

The goal of these comments is to tie the code to the description you have provided here. Please do not alter the actual content of your code - only add comments/docstrings.

**Submitting your documentation and commented code**

Please submit your commented code within a .zip or equivalent file type (1 file per solution), and share it with us as an attachment alongside this Word doc.

You can send these via email (to [nm736@cam.ac.uk](mailto:nm736@cam.ac.uk); [james.jordon@wolfson.ox.ac.uk](mailto:james.jordon@wolfson.ox.ac.uk); [es583@cam.ac.uk](mailto:es583@cam.ac.uk)) or DM James Jordon/Evgeny Saveliev on Slack (you can join the workspace [with this URL](https://join.slack.com/t/hideandseekpr-fbc8582/shared_invite/zt-k2h9xye8-RQNen128uXIG2TRsLa_ppA)).