Trustworthy Hackathon

First Draft

Lucas, Chole, Daivya, Eric

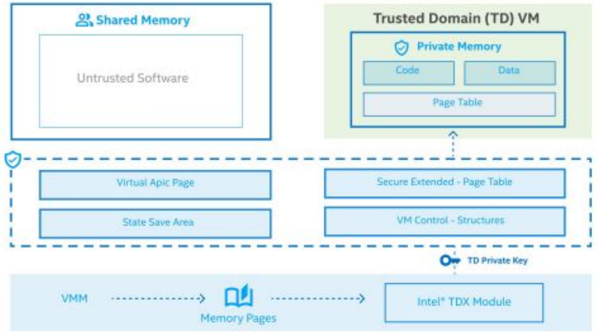
Part 1: Implementation

Technical Analysis

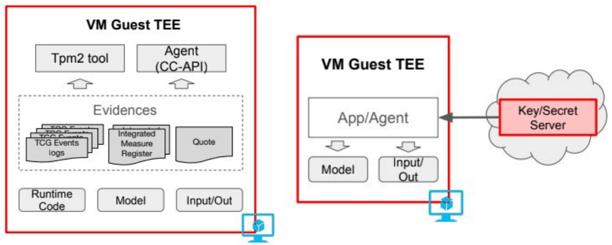
Resources/Technologies

- **Azure Confidential VM -** Provides a secure Ubuntu environment to run operations. Azure Confidential VMs provide numerous security benefits such as Intel TDX and TPM capabilities, isolation and data confidentiality
- **tpm2-tools library -** Allows for attestation utilizing the TPM. Can create endorsement keys to attest for the TPM itself and an attestation key to attest for the evidence. Also provides tools aiding with the modification and evaluation of PCR values and quote creation and verification
- **Azure Key Vault -** Provides a safe remote server to store keys. Can securly transport these keys through the Amazon CLI as they utilize TLS.

Implementation - Please see the markup file in this GitHub Repository for an in depth explanation of the implementation described above
https://github.com/ChloeHouvardas/CC-TPM-Attestation/blob/main/instructions.md



Intel TDX Utilized by Azure's Confidential VMs



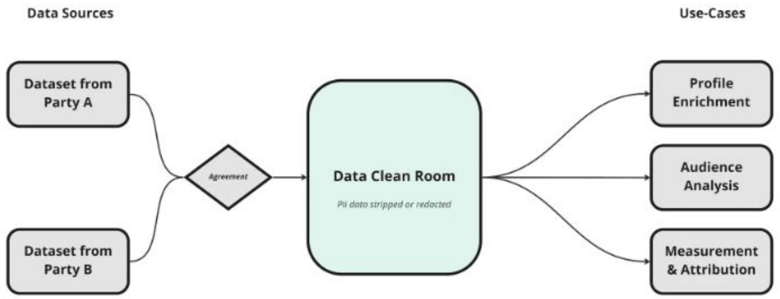
Evidence Collection and Visualization

Protecting Data in Use

Confidential Computing - Currently, there is a vast array of methods to protect data in storage or at rest, but there remains a gap when it comes to protect data in use. Our goal is to navigate an application of confidential computing in a multi-party data sharing scenario.

Goals - Protect data in use by ensuring the confidentiality and integrity of the code and the environment

Purpose - To provide a secure and confidential data sharing platform. In this case, the platform will allow for publishers and advertisers to confidentially combine their data together to extract machine learning insights without having their customer data leaked to the other party or accessed by malicious actors



Part 2: Evaluation

Data cleaning strategy

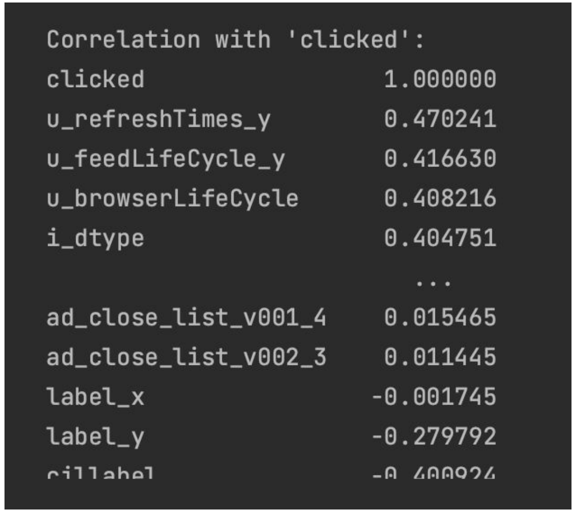
Merging - Merging ads and feeds by variable user_id, combine all the variable from 2 tables and drop some overlapped variables or unnecessary variables and identifiers . E.g. “city”(city is too specific), “e_et”(already have pt_d for time). Then keep the missing data as N/A.(I think this also cause my model to perform badly)

Variable featuring and normalizing - extra month from pt_d by ignore the digits in other position. Separating some long value like 342^2131^124 by “^” and create new columns to put them in.(This is where I get scared, after performing the column division, I created more column, perhaps make the data more complicated) Creating new variable called “clicked” to mark the userid that exist in both table. Sparsing the low frequency value to reduce the number of unique value. Normalizing data to N(0,σ), perform log transformation for highly skewed variable. Convert categorical value to numeric variable

Some conclusion from the dataset

I have some problem with overfitting, I perform logistic regression, random forest, linear regression, SVM, but all seems overfit, this could caused by too many variables and too less data(I only use 1/1000 for training due to memory capacity, so all my conclusion comes from that as well). At this point I cannot provide any information from the machine learning prediction

- The age distribution from whom clicked the ad seems increasing as the age increases until the max point 6-8
- Residence 33 has the highest click volume
- Device 30 has the highest click
- The hour from 7-12 has the highest click/review rate(pretty understandable since it is the entertaining time!)



```
Correlation with 'clicked':
clicked                  1.000000
u_refreshTimes_y         0.470241
u_feedLifeCycle_y        0.416630
u_browserLifeCycle       0.408216
i_dtype                  0.404751
                           ...
ad_close_list_v001_4     0.015465
ad_close_list_v002_3     0.011445
label_x                 -0.001745
label_y                 -0.279792
cillabel                -0.400924
```

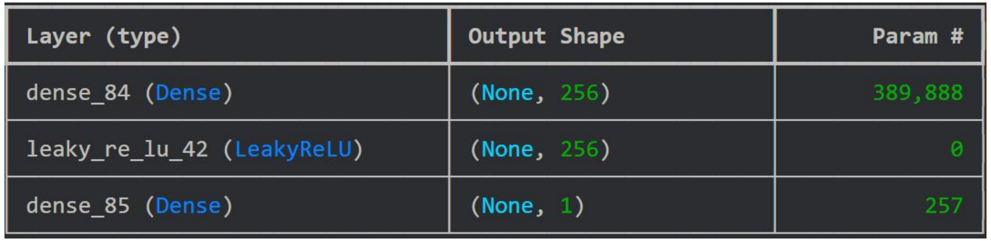
This figure is the correlation figure, notice that 0 is the less related value, negative also means they are related

Part 3: Generating synthetic data

Generating Synthetic Data

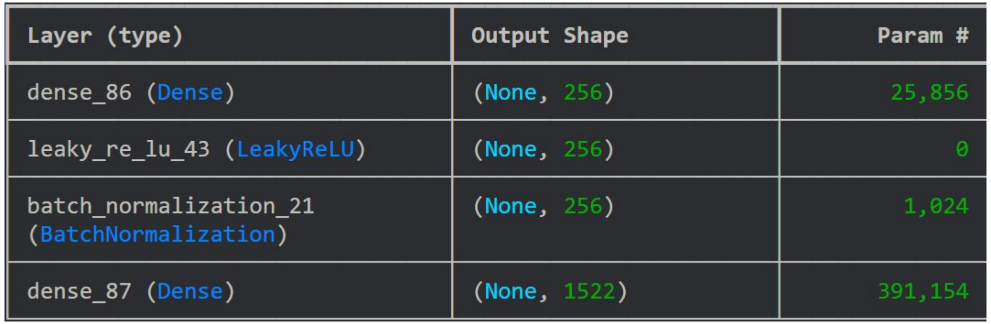
To create our synthetic data for confidentiality purposes a GAN model was used. This model took an input of the provided merged datasets. Due to the large size of the data and issue of using OneHotEncoder to convert categorical data making numerous new columns we had to shrink down the variables. After 400 cycles the Discriminator and Generator both had a loss of ~5% and an accuracy of 99%. From here a dataset of 100k rows was generated with selected columns.

https://github.com/tennyfr/GES-Hackathon

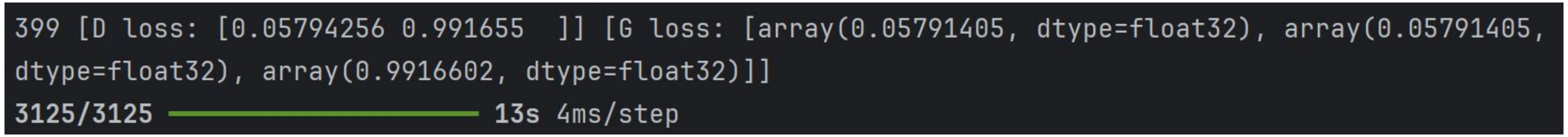


Layer (type)	Output Shape	Param #
dense_84 (Dense)	(None, 256)	389,888
leaky_re_lu_42 (LeakyReLU)	(None, 256)	0
dense_85 (Dense)	(None, 1)	257

Summary of Discriminator ^
and Generator (below)



Layer (type)	Output Shape	Param #
dense_86 (Dense)	(None, 256)	25,856
leaky_re_lu_43 (LeakyReLU)	(None, 256)	0
batch_normalization_21 (BatchNormalization)	(None, 256)	1,024
dense_87 (Dense)	(None, 1522)	391,154



```
399 [D loss: [0.05794256 0.991655  ]] [G loss: [array(0.05791405, dtype=float32), array(0.05791405,
dtype=float32), array(0.9916602, dtype=float32)]]
3125/3125 ━━━━━━━━━━━━━━━━━━━━ 13s 4ms/step
```

Closing Thoughts

There were many restrictions due to the capabilities of our own devices against the size of the large dataset leading to some scaling steps to have to be taken. Overall we have made efforts to set up a Data Clean Room, evaluate the confidential dataset, and generate synthetic data.

Lucas Chin, University of California - Irvine, ‘27

Chloe Houvardas, Queen’s University, ‘27

Chi Lu, University of California - Irvine, ‘25

Daviya

Thank you.