Try before You Buy: Privacy-preserving Evaluation on Cloud-based Machine Learning Data Marketplace

Qiyang Song, Jiahao Cao, Kun Sun, Qi Li, and Xu Ke
What is Data Marketplace?

- A good deep learning model relies on huge good-quality data.
  - Trainers want to enrich their internal data sets with external data.
- As a result, data marketplaces emerge,
  - providing data exchanging platforms for both enterprises and individuals.
Model owners want to **purchase the most valuable data** to improve their models,
but data owners may provide **useless/irrelevant data** that do not improve model performance.
How to evaluate the most valuable data for data shoppers' models?
Cloud needs to access both sellers' data and shoppers' models, but it is untrusted.

Data and models may be sensitive for both sellers and shoppers!

Abuse data and models

Data Owners (Data Sellers) → Pre-train → Cloud → if the model performance is improved, data quality is good → Send Models → Purchase Data → Model Owners (Data Shoppers)

Pre-train
Our Goal:

Provide privacy-preserving ML data evaluation on data marketplaces
Existing Privacy Protection Solutions

- Existing privacy protection solutions
  - Homomorphic Encryption (HE), Secure Multi-party Computation (MPC)
  - Can preserve both the privacy and functionality of data/models on the cloud
- **Limitations**
  - high computational and communication overhead
  - not specially designed for ML data evaluation

We need a lightweight encryption approach that is specially designed for ML data evaluation.
Our Solution

- We design a lightweight encryption approach to protect the privacy of data/models.
  - So, the encrypted data/models cannot be directly evaluated by the cloud.

- We provide a ML evaluation approach that is compatible with our lightweight encryption approach
  - Instead of accessing the original data, we need extra information and mechanisms to evaluate valuable data.
Our System

- Data sellers upload encrypted data to the cloud for sale.

- Data shopper uploads encrypted model and retrieves prediction values to select/validate data.

- The cloud helps the shopper to evaluate sellers' encrypted data.
We need **inner product computation over ciphertexts**.

- For most neural networks, both common matrix and convolution computation can be decomposed to **inner product computation**.
Lightweight ML Encryption

- We use lightweight **inner-product functional encryption (IFE)** and matrix **transformation** to encrypt data/models.
  - Still can use encrypted model to predict/train encrypted data
Lightweight ML Encryption

- We use lightweight inner-product functional encryption (IFE) and matrix transformation to encrypt data/models.
  - Still can use encrypted model to predict/train encrypted data
Lightweight ML Encryption

- We use lightweight **inner-product functional encryption (IFE)** and **matrix transformation** to encrypt data/models.
  - Still can use encrypted model to predict/train encrypted data

![Diagram showing first and second layers](image)
Lightweight ML Encryption

- We use lightweight **inner-product functional encryption (IFE)** and matrix **transformation** to encrypt data/models.
  - Still can use encrypted model to predict/train encrypted data

---

**The first layer**

- Using IFE to encrypt $X$
- Using IFE and matrix transformation to encrypt $W_1$
- $C_1 \times W_1$

**The second layer**

- Matrix Computation
- $C_1 \times Z_1$ where $Z_1 = W_1x$
We use lightweight **inner-product functional encryption (IFE)** and **matrix transformation** to encrypt data/models.

- Still can use encrypted model to predict/train encrypted data

The first layer

- Using IFE to encrypt X

- Using IFE and matrix transformation to encrypt W1

Output: \( C_1^2 \times Z_1^2 \)

Squared activation

Matrix Computation

\( C_1 \times Z_1 \) where \( Z_1 = W_1x \)

The second layer
Lightweight ML Encryption

- We use lightweight inner-product functional encryption (IFE) and matrix transformation to encrypt data/models.
  - Still can use encrypted model to predict/train encrypted data
We use lightweight inner-product functional encryption (IFE) and matrix transformation to encrypt data/models.

- Still can use encrypted model to predict/train encrypted data

**The first layer**

1. Using IFE to encrypt $X$

2. Using IFE and matrix transformation to encrypt $W_1$

**Matrix Computation**

- Output: $C_1^2 \times Z_1^2$

**The second layer**

1. Using matrix transformation to encrypt $W_2$

2. Output: $C_2^2 \times Z_2^2$

**Matrix Computation**

- Output: $C_2 \times Z_2$

- Squared activation

**The first layer**

- $C_1 \times W_1$

- $C_1 \times Z_1$ where $Z_1 = W_1x$

**The second layer**

- $C_2 / C_1^2 \times W_2$

- $C_2 \times Z_2$ where $Z_2 = W_2Z_1^2$
Lightweight ML Encryption

- We use lightweight **inner-product functional encryption (IFE)** and matrix transformation to encrypt data/models.
  - Still can use encrypted model to predict/train encrypted data

The first layer

- Using IFE to encrypt $X$
- Using IFE and matrix transformation to encrypt $W_1$
- Matrix Computation

  $C_1 \times W_1$

  $C_1 \times Z_1$

  where $Z_1 = W_1x$

Output: $C_1^2 \times Z_1^2$

The second layer

- Using matrix transformation to encrypt $W_2$
- Matrix Computation

  $C_2 / C_1^2 \times W_2$

  $C_2 \times Z_2$

  where $Z_2 = W_2Z_1^2$

Output: $C_2^2 \times Z_2^2$

Subsequent layers

Squared activation
Rationale behind Data Selection

- Data selection is based on active learning.
- Active learning uses prediction values (not original data) to evaluate data.
- Valuable data have uncertain prediction values.
  - located near the decision boundary, i.e., provide more information
Data Selection

1. Data sellers and shopper upload their encrypted data and model.
2. The cloud performs prediction operations.
3. Data shopper collects encrypted prediction values to select valuable data.
Data Selection

1. Data sellers and shopper upload their encrypted data and model.
2. The cloud performs prediction operations.
3. Data shopper collects encrypted prediction values to select valuable data.
1. Data sellers and shopper upload their encrypted data and model.
2. The cloud performs prediction operations.
3. Data shopper collects encrypted prediction values to select valuable data.
Data Selection

1. Data sellers and shopper upload their encrypted data and model.
2. The cloud performs prediction operations.
3. Data shopper collects encrypted prediction values to select valuable data.
Data Selection

1. Data sellers and shopper upload their encrypted data and model.
2. The cloud performs prediction operations.
3. Data shopper collects encrypted prediction values to select valuable data.
Data Selection

1. Data sellers and shopper upload their encrypted data and model.
2. The cloud performs prediction operations.
3. Data shopper collects encrypted prediction values to select valuable data.

Select valuable data based on activation learning
Another Problem:

Data selection only considers the informativeness of data, but not labels, not relevance.

What if the selected data contain unintentionally mislabeled data or irrelevant data?
Rationale behind Data Validation

- The shopper and cloud cannot directly see selected data to estimate quality.
- Indirect approach: let the model "try" data and check model performance.
  - "try": use the selected data to retrain the shopper's model.
Rationale behind Data Validation

- The shopper and cloud cannot directly see selected data to estimate quality.
- Indirect approach: let the model "try" data and check model performance.
  - "try": use the selected data to retrain the shopper's model.

![Diagram: Selected data → Re-train → Seller's original model]
Rationale behind Data Validation

- The shopper and cloud **cannot directly see selected data** to estimate quality.
- Indirect approach: let the model "try" data and check model performance.
  - "try" : use the selected data to retrain the shopper's model.
Rationale behind Data Validation

- The shopper and cloud **cannot directly see selected data** to estimate quality.
- Indirect approach: let the model "try" data and check model performance.
  - "try" : use the selected data to retrain the shopper's model.

![Diagram]

- **Selected data** → **Re-train** → **Seller's original model** → **Predict** → **Other data** → **Other data** → **Other data**
- **Retrained model**
- **Prediction values**
  - [0.2, 0.3, ...]
  - [0.3, 0.4, ...]
  - [0.4, 0.5, ...]
Rationale behind Data Validation

- The shopper and cloud **cannot directly see selected data** to estimate quality.
- Indirect approach: let the model "try" data and check model performance.
  - "try" : use the selected data to retrain the shopper's model.

![Diagram showing data validation process]

- **Selected data** → **Re-train** → **Seller's original model** → **Other data** → **Predict** → **Retrained model** → **Prediction values** → **See the prediction performance**
Rationale behind Data Validation

- The shopper and cloud cannot directly see selected data to estimate quality.
- Indirect approach: let the model "try" data and check model performance.
  - "try" : use the selected data to retrain the shopper's model.

![Diagram showing data validation process]

1. Selected data is re-trained with the seller's original model.
2. The re-trained model predicts on other data.
3. Prediction values are compared:
   - If the performance is improved, data quality is good.
   - Prediction values shown: [0.2, 0.3, ...], [0.3, 0.4, ...], [0.4, 0.5, ...]

See the prediction performance.
Data Validation

1. Cloud uses the selected data to retrain the shopper's encrypted model.
2. Cloud uses the retrained model to predict uniformly distributed data.
3. The shopper **collects encrypted prediction values** to estimate data quality.
Data Validation

1. Cloud uses the selected data to retrain the shopper's encrypted model.
2. Cloud uses the retrained model to predict uniformly distributed data.
3. The shopper collects encrypted prediction values to estimate data quality.

Cloud

Data shopper
Data Validation

1. Cloud uses the selected data to retrain the shopper's encrypted model.
2. Cloud uses the retrained model to predict uniformly distributed data.
3. The shopper collects encrypted prediction values to estimate data quality.
Data Validation

1. Cloud uses the selected data to retrain the shopper's encrypted model.
2. Cloud uses the retrained model to predict uniformly distributed data.
3. The shopper collects encrypted prediction values to estimate data quality.
Data Validation

1. Cloud uses the selected data to retrain the shopper's encrypted model.
2. Cloud uses the retrained model to predict uniformly distributed data.
3. The shopper collects encrypted prediction values to estimate data quality.
Data Validation

1. Cloud uses the selected data to retrain the shopper's encrypted model.
2. Cloud uses the retrained model to predict uniformly distributed data.
3. The shopper collects encrypted prediction values to estimate data quality.
Experiment Setup

- We simulate
  - 100 sellers and 1 shopper
  - divide MINIST to 101 subsets, assigned to sellers and shopper

- We evaluate
  - benefits of our data selection
  - the accuracy of our data validation
  - computational overhead
Benefits of Our Data Selection

- Compared with random selection, our data selection can reduce about 60% prediction errors.

* Model 1 and 2 are trained with 5500 and 55000 samples, respectively.
Simulate low-quality samples that are most likely to evade data validation

* We split samples into multiple subsets and validate them one by one.
* Validation granularity means that the size of validation subsets.
Computational Overhead

- Compared with homomorphic encryption based approach (E2DM)

Table 1: Execution Time of CNN models

<table>
<thead>
<tr>
<th>Operations</th>
<th>Execution Time (second)</th>
<th>E2DM</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Encryption</td>
<td>0.40</td>
<td>0.48</td>
<td></td>
</tr>
<tr>
<td>Model Encryption</td>
<td>0.14</td>
<td>0.20</td>
<td></td>
</tr>
<tr>
<td>Feed Forward</td>
<td>35.88</td>
<td>2.59</td>
<td></td>
</tr>
<tr>
<td>Back Propagation</td>
<td>N/A</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

* We encrypt a six-layer CNN model and measure relevant operations
Conclusion

● A privacy-preserving and efficient ML data evaluation framework on data marketplaces

● A new lightweight ML encryption protocol that can preserve both privacy and functionality of data/models on the cloud
  ➢ Based on IFE and matrix transformation

● Privacy-preserving Data Selection and Validation
  ➢ Can select valuable data and validate the data quality
  ➢ Do not disclose the original data and models
Thank you!
Backup: IFE-based Matrix Encryption

- We can use **inner product functional encryption** to enable matrix or convolution computation over ciphertexts.

  - The result is plaintext, we apply **matrix transformation** to hide the result.
• IFE is only used to encrypt the first layer since it only support simple inner product computation.
• Remaining layers are encrypted by matrix transformation (see our paper).
During prediction, the output of each layer is $C_i^2 \times Z_i^2$ ($Z_i^2$ is original output).

We can decrypt the output by multiply $C_i^{-2}$.

**The first layer**

Using IFE to encrypt

$X$

Using IFE and mat transformation to encrypt

$C1 \times W1$

output: $C1^2 \times Z1^2$

Squared activation

Mat Computation

$C1 \times Z1$ where $Z1 = W1X$

**The second layer**

Using matrix transformation to encrypt

$C2 / C1^2 \times W2$

output: $C2^2 \times Z2^2$

Squared activation

Mat Computation

$C2 \times Z2$ where $Z2 = W2Z1^2$
To evaluate data of different values, we set a variable threshold $T$. $T$ is often the previous prediction errors. If the current prediction errors $< T$, we can say the performance is improved, and the data quality is good.
Backup: TEE

- TEE may leak some sensitive information.
  - Cache Attacks
  - Fault injection attacks
- TEE has some memory limits.
  - For SGX, 128 MB