Label Inference Attacks Against Vertical Federated Learning

Chong Fu    Xuhong Zhang    Shouling Ji    Jinyin Chen    Jingzheng Wu    Shanqing Guo    Jun Zhou    Alex X. Liu    Ting Wang

2021
Big Data Era

- Locations
- Health records
- View histories
- ...

Private user data

Data collection

IT companies’
- Apps
- Websites
- ...

Data analysis (includes machine learning)
Data Leakage

2018, Facebook exposed 87 million user data

2019, Capital One Bank leaked 106 million user data

2020, Marriott Hotel breached 5.2 million user data

2020, Brazilian ministry of health leaked 0.24 billion records

2020, Microsoft exposed 250 million records

2020, 6.4 million voters’ data in Israel were leaked
Traditional **centralized** machine learning breaks laws of user data protection.
The Dilemma of “Isolated Data”

- The dilemma of “isolated data”

User data is isolated in different companies or organizations.
Federated Learning

- The dilemma of “isolated data”
- Federated learning (FL)

Traditional **centralized** machine learning breaks laws of user data protection.

FL allows multiple participants to collaboratively train a machine learning model **without revealing** their local data.
Horizontal federated learning (HFL):
Datasets share the **same feature space** but differ in the **sample space**.

Vertical federated learning (VFL):
Datasets share the **same sample space** but **differ in the feature space**.
Federated Learning Is Widely Used

- FL is being widely used in industry. Worldwide IT companies put much effort into developing FL systems.

- TensorFlow Federated from Google
- PySyft from OpenMined

- Federated AI Technology Enabler (FATE) from Tencent
- Fedlearner from ByteDance
- PaddleFL from Baidu
Federated Learning Has Vulnerabilities

An adversarial participant in federated learning may:

**Infer private information of other participants**
- Infer membership [Oakland’ 19]
- Infer class representatives [CCS’ 17]
- Infer sample properties [Oakland’ 19]
- Reconstructing training samples [NeurIPS’ 19]
- ...

**Attack the federated model**
- Inject backdoor to the federated model [ICLR’ 20]
- Poison the federated model [USENIX Security’ 20]
- ...

- Above studies have thoroughly analyzed the privacy and security risks of HFL. **However, the privacy risks of VFL remain unexplored.**
- We reveal and shed lights on the vulnerability of VFL to the **label inference attacks.**
Label Inference Attacks
Illustration of Label Inference Attacks Against VFL with Model Splitting

- Several participants collaboratively train a VFL model.

Gradients of the loss w.r.t. outputs of the bottom model
Several participants collaboratively train a VFL model.

Every participant in VFL holds partial features.
Several participants collaboratively train a VFL model.

Every participant in VFL holds **partial features**.

The **labels** are privately owned by one participant. This participant also controls the server running the **top model**.
Several participants collaboratively train a VFL model.

Every participant in VFL holds **partial features**.

The **labels** are privately owned by one participant. This participant also controls the server running the **top model**.

One of the participants without labels is the **adversary**, whose goal is to infer the privately owned labels.
Attack 1: Passive Label Inference Attack

- Exploit the locally **owned bottom model**.
Attack 1: Passive Label Inference Attack

- Exploit the locally **owned bottom model**.
- Fine-tune the bottom model with an **additional classification layer**.

![Diagram](image)
Attack 1: Passive Label Inference Attack

- Exploit the locally **owned bottom model**.
- Complete the bottom model with an **additional classification layer**.
- Use a small amount of auxiliary labeled data to **fine-tune** the bottom model in a **semi-supervised** manner.

![Diagram of attack的过程中](image_url)

1. LocalOptimizer
2. (Trained) Bottom Model
3. Bottom Model
4. Semi-supervised learning loss
5. Inferred labels
Attack 2: Active Label Inference Attack

- **Accelerate** the local model’s learning during training

---

**Algorithm 1** Local malicious optimization of the adversary’s bottom model

**Require:** Momentum parameter $\beta$, the gradient scaling factor’s resetting parameter $\gamma$, maximum gradient scaling factor $r_{\text{max}}$, minimum gradient scaling factor $r_{\text{min}}$, learning rate $\eta$, initial bottom model parameters $\Theta$, initial gradient velocity $v$.

1: **while** stopping criterion not met **do**
2: Receive $G_{\text{out put}}$ from the server
3: $G \leftarrow \text{Backward}(G_{\text{out put}})$
4: **for** each parameter $\theta$ in $\Theta$ and its gradient $g_\theta$ in $G$ **do**
5: $v_\theta \leftarrow \beta \cdot v_\theta + (1 - \beta) \cdot g_\theta$
6: **if** is not the first criterion **then**
7: $r_\theta \leftarrow 1.0 + \gamma \cdot (v_\theta \div v_{\text{last}})$
8: $r_\theta \leftarrow \text{Max}(r_\theta, r_{\text{min}})$
9: $r_\theta \leftarrow \text{Min}(r_\theta, r_{\text{max}})$
10: $v_\theta \leftarrow r_\theta \cdot v_{\text{last}}$
11: **end if**
12: $v_{\text{last}} \leftarrow v_\theta$
13: $\Theta \leftarrow \Theta - \eta \cdot v_\theta$
14: **end for**
15: **end while**
Attack 2: Active Label Inference Attack

- **Accelerate** the local model’s learning during training
- Better expressiveness of the bottom model
- The VFL model is tricked to rely more on the adversary’s bottom model

**Algorithm 1** Local malicious optimization of the adversary’s bottom model

```
Require: 
Momentum parameter $\beta$, the gradient scaling factor’s resetting parameter $\gamma$, maximum gradient scaling factor $r_{\text{max}}$, minimum gradient scaling factor $r_{\text{min}}$, learning rate $\eta$, initial bottom model parameters $\Theta$, initial gradient velocity $v$.
1: while stopping criterion not met do
2:   Receive $G_{\text{output}}$ from the server
3:   $G \leftarrow \text{Backward}(G_{\text{output}})$
4:   for each parameter $\theta$ in $\Theta$ and its gradient $g_{\theta}$ in $G$ do
5:     $v_{\theta} \leftarrow \beta \cdot v_{\theta} + (1 - \beta) \cdot g_{\theta}$
6:     if is not the first criterion then
7:       $r_{\theta} \leftarrow 1.0 + \gamma \cdot (v_{\theta} \div v_{\text{last}})$
8:       $r_{\theta} \leftarrow \text{Max}(r_{\theta}, r_{\text{min}})$
9:       $r_{\theta} \leftarrow \text{Min}(r_{\theta}, r_{\text{max}})$
10:      $v_{\theta} \leftarrow r_{\theta} \cdot v_{\text{last}}$
11:   end if
12:   $v_{\text{last}} \leftarrow v_{\theta}$
13:   $\theta \leftarrow \theta - \eta \cdot v_{\theta}$
14: end for
15: end while
```
Several participants collaboratively train a VFL model.

During the forward propagation, this participant sums up outputs of all the bottom models to get the final output.
Several participants collaboratively train a VFL model. During the forward propagation, this participant **sums up outputs of all the bottom models** to get the final output.

- Every participant holds **partial features**.
- The **labels** are privately owned by one participant.
Several participants collaboratively train a VFL model.

During the forward propagation, this participant sums up outputs of all the bottom models to get the final output.

Every participant holds partial features.

The labels are privately owned by one participant.

One of the participants without labels is the adversary, whose goal is to infer the privately owned labels.
For VFL without model splitting, the adversary is able to receive the **gradients of the final prediction layer**.

**Attack 3: Direct Label Inference Attack**

- Sum up outputs of all bottom models
- Gradients of the loss w.r.t. outputs of the bottom model
For VFL without model splitting, the adversary is able to receive the **gradients of the final prediction layer**.

Directly infer labels by analyzing the **signs of the gradients** received from the server.

\[
loss(x, c) = -\log \frac{e^{\sum_k x_k}}{\sum_j e^{\sum_i y_j^i}}
\]

\[
\frac{\partial{loss(x, c)}}{\partial{y_i^{adv}}} = \frac{-\frac{\partial}{\partial{x_i}} \log(e^{\sum_j y_j^i})}{\sum_j e^{\sum_i y_j^i}}
= \begin{cases} 
-1 + \frac{1}{\sum_j e^{y_j^i}} & \text{if } i = c \\
\frac{e^{y_i^i}}{\sum_j e^{y_j^i}} & \text{if } i \neq c 
\end{cases}
\]

Sum up outputs of all bottom models

Gradients of the loss w.r.t. outputs of the bottom model

Output

Label

Participant A

Feature A

Bottom Model A

Participant B

Feature B

Bottom Model B

Feature B

Label
Attack Evaluation
Experimental Setup

Datasets and model architectures

- **Various data types**
  - Image, text, numerical feature and categorical feature.

- **Various model architectures**

- The VFL models get **good performance on the original tasks**.
  - **Top-1 accuracy:**
    - CIFAR-10: 82.80%
    - CINIC-10: 73.69%
    - Yahoo Answers: 71.67%
    - Criteo: 71.32%
  - **Top-5 accuracy on CIFAR-100:** 75.11%
  - **F1 score on BHI:** 83.40%
Experimental Setup

Datasets and model architectures

- **Various data types**
  Image, text, numerical feature and categorical feature.

- **Various model architectures**

- The VFL models get **good performance on the original tasks**.
  - Top-1 accuracy:
    - CIFAR-10: 82.80%
    - CINIC-10: 73.69%
    - Yahoo Answers: 71.67%
    - Criteo: 71.32%
  - Top-5 accuracy on CIFAR-100: 75.11%
  - F1 score on BHI: 83.40%
Datasets and model architectures

- **Various data types**
  - Image, text, numerical feature and categorical feature.

- **Various model architectures**

- **The VFL models get good performance on the original tasks.**
  - **Top-1 accuracy:**
    - CIFAR-10: 82.80%
    - CINIC-10: 73.69%
    - Yahoo Answers: 71.67%
    - Criteo: 71.32%
  - **Top-5 accuracy on CIFAR-100:** 75.11%
  - **F1 score on BHI:** 83.40%

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Bottom Model Architecture</th>
<th>Top Model Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>ResNet-18</td>
<td>FCNN-4</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>ResNet-18</td>
<td>FCNN-4</td>
</tr>
<tr>
<td>CINIC-10</td>
<td>ResNet-18</td>
<td>FCNN-4</td>
</tr>
<tr>
<td>Yahoo Answers</td>
<td>BERT</td>
<td>FCNN-4</td>
</tr>
<tr>
<td>Criteo</td>
<td>FCNN-3</td>
<td>FCNN-3</td>
</tr>
<tr>
<td>BHI</td>
<td>ResNet-18</td>
<td>FCNN-4</td>
</tr>
</tbody>
</table>
Performance of the Passive/Active/Direct Label Inference Attack

- Good attack performance.
- The active label inference attack outperforms the passive label inference attack.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train Set Size</th>
<th>Test Set Size</th>
<th>Number of Classes</th>
<th>Known Label Quantity Per Class</th>
<th>Metric</th>
<th>Attack Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Train Set</td>
<td>Passive</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>50,000</td>
<td>10,000</td>
<td>10</td>
<td>4</td>
<td>Top-1 Acc</td>
<td>0.8024</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>50,000</td>
<td>10,000</td>
<td>100</td>
<td>4</td>
<td>Top-5 Acc</td>
<td>0.6267</td>
</tr>
<tr>
<td>CINIC-10</td>
<td>180,000</td>
<td>90,000</td>
<td>10</td>
<td>4</td>
<td>Top-1 Acc</td>
<td>0.7206</td>
</tr>
<tr>
<td>Yahoo Answers</td>
<td>50,000</td>
<td>20,000</td>
<td>10</td>
<td>10</td>
<td>Top-1 Acc</td>
<td>0.6335</td>
</tr>
<tr>
<td>Criteo</td>
<td>80,000</td>
<td>20,000</td>
<td>2</td>
<td>50</td>
<td>Top-1 Acc</td>
<td>0.6828</td>
</tr>
<tr>
<td>BHI</td>
<td>69,181</td>
<td>17,296</td>
<td>2</td>
<td>35</td>
<td>F1 Score</td>
<td>0.7614</td>
</tr>
</tbody>
</table>

- The direct label inference attack can infer all labels in the training dataset (100% top-1 accuracy).
Good attack performance.

The active label inference attack outperforms the passive label inference attack.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train Set Size</th>
<th>Test Set Size</th>
<th>Number of Classes</th>
<th>Known Label Quantity Per Class</th>
<th>Metric</th>
<th>Attack Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Train Set</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>50,000</td>
<td>10,000</td>
<td>10</td>
<td>4</td>
<td>Top-1 Acc</td>
<td>0.8024</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>50,000</td>
<td>10,000</td>
<td>100</td>
<td>4</td>
<td>Top-5 Acc</td>
<td>0.6267</td>
</tr>
<tr>
<td>CINIC-10</td>
<td>180,000</td>
<td>90,000</td>
<td>10</td>
<td>4</td>
<td>Top-1 Acc</td>
<td>0.7206</td>
</tr>
<tr>
<td>Yahoo Answers</td>
<td>50,000</td>
<td>20,000</td>
<td>10</td>
<td>10</td>
<td>Top-1 Acc</td>
<td>0.6335</td>
</tr>
<tr>
<td>Criteo</td>
<td>80,000</td>
<td>20,000</td>
<td>2</td>
<td>50</td>
<td>Top-1 Acc</td>
<td>0.6828</td>
</tr>
<tr>
<td>BHI</td>
<td>69,181</td>
<td>17,296</td>
<td>2</td>
<td>35</td>
<td>F1 Score</td>
<td>0.7614</td>
</tr>
</tbody>
</table>

The direct label inference attack can infer all labels in the training dataset (100% top-1 accuracy).
Performance of the Passive/Active/Direct Label Inference Attack

- Good attack performance.
- The active label inference attack outperforms the passive label inference attack.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train Set Size</th>
<th>Test Set Size</th>
<th>Number of Classes</th>
<th>Known Label Quantity Per Class</th>
<th>Metric</th>
<th>Attack Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>50,000</td>
<td>10,000</td>
<td>10</td>
<td>4</td>
<td>Top-1 Acc</td>
<td>0.8024 0.8484</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>50,000</td>
<td>10,000</td>
<td>100</td>
<td>4</td>
<td>Top-5 Acc</td>
<td>0.6267 0.6732</td>
</tr>
<tr>
<td>CINIC-10</td>
<td>180,000</td>
<td>90,000</td>
<td>10</td>
<td>4</td>
<td>Top-1 Acc</td>
<td>0.7206 0.7818</td>
</tr>
<tr>
<td>Yahoo Answers</td>
<td>50,000</td>
<td>20,000</td>
<td>10</td>
<td>10</td>
<td>Top-1 Acc</td>
<td>0.6335 0.6424</td>
</tr>
<tr>
<td>Criteo</td>
<td>80,000</td>
<td>20,000</td>
<td>2</td>
<td>50</td>
<td>Top-1 Acc</td>
<td>0.6828 0.6879</td>
</tr>
<tr>
<td>BHI</td>
<td>69,181</td>
<td>17,296</td>
<td>2</td>
<td>35</td>
<td>F1 Score</td>
<td>0.7614 0.7824</td>
</tr>
</tbody>
</table>

- The direct label inference attack can infer all labels in the training dataset (100% top-1 accuracy).
More auxiliary labeled samples indeed increases the attack accuracy.

However, as the number of auxiliary labeled samples grows, the attack accuracy increases more and more slowly.

The trained bottom model contains much information for label inference.

<table>
<thead>
<tr>
<th>Known Label Quantity</th>
<th>Passive Label Inference</th>
<th>Direct Semi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Dataset</td>
<td>Test Dataset</td>
</tr>
<tr>
<td>10</td>
<td>0.6554</td>
<td>0.5235</td>
</tr>
<tr>
<td>20</td>
<td>0.7080</td>
<td>0.5542</td>
</tr>
<tr>
<td>40</td>
<td>0.8024</td>
<td>0.6299</td>
</tr>
<tr>
<td>120</td>
<td>0.8406</td>
<td>0.6305</td>
</tr>
<tr>
<td>320</td>
<td>0.8544</td>
<td>0.6392</td>
</tr>
</tbody>
</table>

Experiment on CIFAR-10. Attack performance is measured by top-1 accuracy.
More auxiliary labeled samples indeed increases the attack accuracy.

However, as the number of auxiliary labeled samples grows, the attack accuracy increases more and more slowly.

The trained bottom model contains much information for label inference.

<table>
<thead>
<tr>
<th>Known Label Quantity</th>
<th>Passive Label Inference</th>
<th>Direct Semi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Dataset</td>
<td>Test Dataset</td>
</tr>
<tr>
<td>10</td>
<td>0.6554</td>
<td>0.5235</td>
</tr>
<tr>
<td>20</td>
<td>0.7080</td>
<td>0.5542</td>
</tr>
<tr>
<td>40</td>
<td>0.8024</td>
<td>0.6299</td>
</tr>
<tr>
<td>120</td>
<td>0.8406</td>
<td>0.6305</td>
</tr>
<tr>
<td>320</td>
<td>0.8544</td>
<td>0.6392</td>
</tr>
</tbody>
</table>

Experiment on CIFAR-10. Attack performance is measured by top-1 accuracy.
More auxiliary labeled samples indeed increases the attack accuracy.

However, as the number of auxiliary labeled samples grows, the attack accuracy increases more and more slowly.

The trained bottom model contains much information for label inference.

<table>
<thead>
<tr>
<th>Known Label Quantity</th>
<th>Passive Label Inference</th>
<th>Direct Semi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Dataset</td>
<td>Test Dataset</td>
</tr>
<tr>
<td>10</td>
<td>0.6554</td>
<td>0.5235</td>
</tr>
<tr>
<td>20</td>
<td>0.7080</td>
<td>0.5542</td>
</tr>
<tr>
<td>40</td>
<td>0.8024</td>
<td>0.6299</td>
</tr>
<tr>
<td>120</td>
<td>0.8406</td>
<td>0.6305</td>
</tr>
<tr>
<td>320</td>
<td>0.8544</td>
<td>0.6392</td>
</tr>
</tbody>
</table>

Experiment on CIFAR-10. Attack performance is measured by top-1 accuracy.
The active label inference attack has a very small impact on the federated model's performance on the original task.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>Model Performance under:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>No Attack</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>Top-1 Acc</td>
<td>0.8280</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>Top-5 Acc</td>
<td>0.7511</td>
</tr>
<tr>
<td>CINIC-10</td>
<td>Top-1 Acc</td>
<td>0.7369</td>
</tr>
<tr>
<td>Yahoo Answers</td>
<td>Top-1 Acc</td>
<td>0.7167</td>
</tr>
<tr>
<td>Criteo</td>
<td>Top-1 Acc</td>
<td>0.7132</td>
</tr>
<tr>
<td>BHI</td>
<td>F1 Score</td>
<td>0.8340</td>
</tr>
</tbody>
</table>
The quantity of the adversary’s local features determines the upper bound of the attack performance.

The active label inference attack can only boost the attack performance within this upper bound.

The impact of the quantity of the adversary’s features on Criteo. The upper bound is obtained using all the labels to directly train an inference model with the adversary’s features.
The more layers the adversary's bottom model has, the better the active label inference attack performs.

VFL models with simpler tasks face a greater risk of label leakage.
Performance of the Active Attack in Multi-party Setting

- Attack performance degrades as the number of participants increases.
- Label inference attacks threaten multi-party VFL even when there are 8 participants.

Figure 3: Performance of the active attack in multi-party setting on BHI.
Performance of the Active Attack in Multi-party Setting

- Attack performance degrades as the number of participants increases.
- Label inference attacks threaten multi-party VFL even when there are 8 participants.

Figure 3: Performance of the active attack in multi-party setting on BHI.
Why the Active Label Inference Attack Works (1)

More attention is drawn to the adversary’s datum under the active attack.

GradCAM visualization of some training samples under the passive or active label inference attacks on CIFAR-10. The left half of the image is the datum of the adversary.
The adversary’s bottom model learns **better representations of raw local data** under the active attack.
Defense Evaluation
Possible Defense

- In the training process of VFL, the **only** information sent to the adversary is the **gradients from the server**.
- Defense strategies can be applied to the gradients to prevent information leakage from the server to the adversary.
In the training process of VFL, the **only** information sent to the adversary is the **gradients from the server**.

Defense strategies can be applied to the gradients to prevent information leakage from the server to the adversary.

- Prune small gradients
- Add noise
- Discretization
Defense Against the Active Label Inference Attack

- Three mainstream defense approaches: noisy gradients, gradient compression and privacy-preserving deep learning.
- DiscreteSGD, a customized version of the defense approach signSGD.
- These defense approaches are not effective against our active label inference attack.
Two of the four evaluated defense approaches can successfully mitigate the direct label inference attack.

Table 6: Defenses against the direct label inference attack on CIFAR-10.

<table>
<thead>
<tr>
<th>Defense Approach</th>
<th>Parameter</th>
<th>Parameter Set Value</th>
<th>Model Accuracy</th>
<th>Attack Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy Gradients</td>
<td>Noise Scale</td>
<td>1e-4</td>
<td>0.8347</td>
<td>0.8063</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-3</td>
<td>0.8318</td>
<td>0.4906</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-2</td>
<td>0.7191</td>
<td>0.2432</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1e-1</td>
<td>0.1000</td>
<td>0.1265</td>
</tr>
<tr>
<td>Gradient Compression</td>
<td>Compression Rate</td>
<td>75%</td>
<td>0.8248</td>
<td>0.9997</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50%</td>
<td>0.8259</td>
<td>0.9931</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25%</td>
<td>0.8049</td>
<td>0.9245</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10%</td>
<td>0.1000</td>
<td>0.0058</td>
</tr>
<tr>
<td>Privacy-preserving Deep Learning</td>
<td>$\theta_{\mu}$</td>
<td>0.75</td>
<td>0.8189</td>
<td>0.3904</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.50</td>
<td>0.8216</td>
<td>0.3891</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.25</td>
<td>0.1993</td>
<td>0.0972</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.10</td>
<td>0.1000</td>
<td>0.0430</td>
</tr>
<tr>
<td>Discrete SGD</td>
<td>N</td>
<td>24</td>
<td>0.8145</td>
<td>0.9763</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18</td>
<td>0.7962</td>
<td>0.9330</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>0.7471</td>
<td>0.9399</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>0.6575</td>
<td>0.9087</td>
</tr>
</tbody>
</table>
Conclusion
We reveal and shed lights on the **new label leakage issue of VFL**.

We present **three types of label inference attacks against VFL**. We evaluate our attacks on various tasks under both two-participant and multi-participant settings and achieve **good attack performance**.

We share **insights about the underlying working mechanism** of the active label inference attack, and present visualized proofs.

We evaluate four possible defenses against our attacks and find that they are not effective against the passive/active attack, which **motivates future work on better defenses**.