USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

# Suda: An Efficient and Secure Unbalanced Data Alignment Framework for Vertical Privacy-Preserving Machine Learning

Lushan Song, *Fudan University and ByteDance;* Qizhi Zhang and Yu Lin, *ByteDance;* Haoyu Niu, *Fudan University;* Daode Zhang, *ByteDance;* Zheng Qu and Weili Han, *Fudan University;* Jue Hong, Quanwei Cai, and Ye Wu, *ByteDance*

https://www.usenix.org/conference/usenixsecurity25/presentation/song-lushan

This artifact appendix is included in the Artifact Appendices to the Proceedings of the 34th USENIX Security Symposium and appends to the paper of the same name that appears in the Proceedings of the 34th USENIX Security Symposium.

August 13–15, 2025 • Seattle, WA, USA

978-1-939133-52-6

Open access to the Artifact Appendices to the Proceedings of the 34th USENIX Security Symposium is sponsored by USENIX.


# USENIX Security '25 Artifact Appendix: Suda: An Efficient and Secure Unbalanced Data Alignment Framework for Vertical Privacy-Preserving Machine Learning

Lushan Song
Fudan University & ByteDance

Qizhi Zhang
ByteDance

Yu Lin
ByteDance

Haoyu Niu
Fudan University

Daode Zhang
ByteDance

Zheng Qu
Fudan University

Weili Han
Fudan University

Jue Hong
ByteDance

Quanwei Cai
ByteDance

Ye Wu
ByteDance

## A Artifact Appendix

### A.1 Abstract

In this paper, we propose `Suda`, an efficient and secure unbalanced data alignment framework for vertical privacy-preserving machine learning (VPPML). `Suda` efficiently, directly, and exclusively outputs data shares in the intersection without expensive secure shuffle operations. Consequently, `Suda` efficiently and seamlessly aligns with secure training in VPPML.

This artifact is a C++ implementation of the protocols presented in this paper. It contains the source code and scripts needed to reproduce our experimental results in this paper. Specifically, we provide: (1) Source code for the functionalities of the secure unbalanced data alignment and batch privacy information retrieve (PIR). (2) Scripts for obtaining the experimental results in our paper.

### A.2 Description & Requirements

`Suda` enables a server $P_S$ that holds larger data with size $N$ and a client that holds smaller data with size $n$ ($n \ll N$ ) to achieve secure data alignment efficiently. After executing `Suda`, $P_S$ and $P_C$ can obtain data shares in the intersection without data shares outside the intersection. Then, these two parties could use the data shares as input to train a machine learning model securely.

In addition, we provide an efficient implementation of the batch PIR protocol. This protocol enables a client $P_C$ to retrieve results corresponding to its batch of queries from the data of a server $P_S$.

#### A.2.1 Security, privacy, and ethical concerns

There are no security, privacy, or ethical concerns associated with this artifact.

#### A.2.2 How to access

The artifact is open-sourced both at Zenodo https://zenodo.org/records/15109398 and in the Git repository https://github.com/sls33/Suda.

#### A.2.3 Hardware dependencies

Our experiments were run on a Linux server equipped with an Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz and 720GB of RAM. Please make sure that the machine has at least 500 GB of RAM to obtain the experimental results in our paper.

#### A.2.4 Software dependencies

We recommend you install the following software dependencies:

- Debian GNU/Linux 10 (buster) or greater.
- Python 3.7 or greater.
- gcc 11.5.0 or greater.
- clang 18.1.8 or greater.

#### A.2.5 Benchmarks

We employ two datasets as follows to evaluate `Suda`'s performance over public datasets.

- SVHN[1], which contains 73257 training samples and 26032 test samples, each represented as a 32 × 32 RGB image.

[1] http://ufldl.stanford.edu/housenumbers/

- Character Font Images[2], which contains 832670 samples, for each sample we select a $20 \times 20$ grayscale image and 8 additional features.

Besides, we use logistic regression to evaluate the performance of secure training using the outputs of secure unbalanced data alignment.

## A.3 Set-up

Follow the "Requirement" of README documentation in the Git repository https://github.com/sls33/Suda, which will guide you through setting up the required workspace.

### A.3.1 Installation

We provide instructions on how to install the dependencies and necessary configuration steps in the README documentation of https://github.com/sls33/Suda.

### A.3.2 Basic Test

After installing all the dependencies and third-party libraries, you can run the command `$ ./build/bin/psi_to_share_test 20 1024 100 0 test_ps.txt & ./build/bin/psi_to_share_test 20 1024 100 1 test_pc.txt` to run a simple functionality test. In this command, 20 refers to the larger data size $N = 2^{20}$, 1024 refers to the smaller data size $n = 1024$, and 100 refers to the feature dimensions $m = 100$. The results are stored in the files "test_ps.txt" and "test_pc.txt" as follows.
test_ps.txt:

```
host_log_n_data=20
batch_size=1024
feature_num=100
party_id=0
Performance:
communication size (send + recv): 91951.4 KBytes
mem usage of server: 2317.16MB
total time of server: 191306
```

test_pc.txt:

```
host_log_n_data=20
batch_size=1024
feature_num=100
party_id=1
Performance:
communication size (send + recv): 91951.4 KBytes
mem usage of client: 293.742MB
total time of client: 191728
```

[2] https://archive.ics.uci.edu/dataset/417/character+font+images

## A.4 Evaluation workflow

### A.4.1 Major Claims

**(C1):** Suda can efficiently achieve secure unbalanced data alignment between two parties. This is proven by the experiment (E1). The experimental results are described in Section 5.2 of the paper and illustrated in Tables 1, 2, 3, and 4.
**(C2):** Suda's outputs enhance the efficiency of secure training in VPPML. Therefore, Suda seamlessly aligns secure training in VPPML. This is proven by the experiment (E2). The experimental results are described in Section 5.2.1 of the paper and illustrated in Table 1.
**(C3):** Suda can efficiently achieve batch PIR. This is proven by the experiment (E3). The experimental results are described in Section 5.3 of the paper and illustrated in Table 5.

### A.4.2 Experiments

**(E1):** *[Efficiency of Secure Unbalanced Data Alignment] [30 human-minutes + 25 compute-hour + 100GB disk + 300GB RAM]: This experiment evaluates the efficiency of secure unbalanced data alignment in Suda over different data settings, including public datasets, varied data sizes, varied feature dimensions and varied intersection size.*
**Preparation:** Build the environment as described in the following link https://github.com/sls33/Suda.
**Execution:** Follow the scripts of "Efficiency of secure unbalanced data alignment" in the following link https://github.com/sls33/Suda to run the experiments and retrieve the results. Note that the source code of the baseline CPSI is in https://github.com/Visa-Research/volepsi.git. If you want to obtain their experiment results, you can follow the instructions in their repository.
**Results:** The experimental results would be stored in text files. The README documentation of https://github.com/sls33/Suda provides more details. This experiment supports claim (C1).
**(E2):** *[Efficiency of Secure Training] [10 human-minutes + 10 compute-hour + 100GB disk + 16GB RAM]: This experiment evaluates the efficiency of secure training using the outputs of secure unbalanced training alignment in Suda.*
**Preparation:** Build the environment as described in the following link https://github.com/sls33/Suda.
**Execution:** Follow the scripts of "Efficiency of secure training" in the following link https://github.com/sls33/Suda to run the experiments and retrieve the results.
**Results:** The experimental results would be stored in text files. The README documentation of https://

github.com/sls33/Suda provides more details. This experiment supports claim (C2).

**(E3):** *[Efficiency of Batch PIR] [10 human-minutes + 20 compute-hour + 100GB disk + 500GB RAM]: This experiment evaluates the efficiency of batch PIR protocol in* `Suda`.

**Preparation:** Build the environment as described in the following link https://github.com/sls33/Suda.

**Execution:** Follow the scripts of "Efficiency of batch PIR" in the following link https://github.com/sls33/Suda to run the experiments and retrieve the results. Note that the source code of the baseline `PIRANA` is in https://github.com/zju-abclab/PIRANA. If you want to obtain their experiment results, you can follow the instructions in their repository.

**Results:** The experimental results would be stored in text files. The README documentation of https://github.com/sls33/Suda provides more details. This experiment supports claim (C3).

## A.5 Notes on Reusability

You can configure the larger data size $N$, the smaller data size $n$, and the feature dimensions $m$ in the command as shown in Section A.3.2.

## A.6 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2025/.