Leveraging Crowdsourcing to Detect Improper Tasks in Crowdsourcing Marketplaces

Yukino Baba, Hisashi Kashima (Univ. Tokyo),
Kei Kinoshita, Goushi Yamaguchi,
Yosuke Akiyochi (Lancers Inc.)

IAAI 2013 (July 17, 2013).

Overview: Crowdsourced labels are useful for detecting improper tasks in crowdsourcing

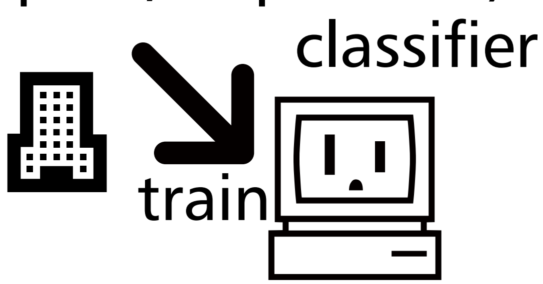
TARGET

Improper task detection in crowdsourcing marketplaces

APPROACH Supervised learning

RESULTS

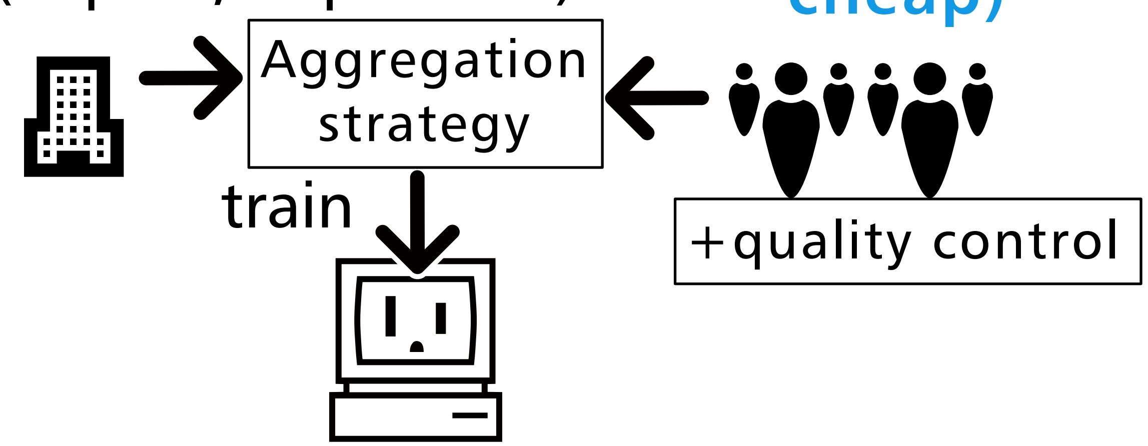
1 Operator (expert, expensive)



Classifier trained using expert labels achieved **AUC 0.950**

2 Operator (expert, expensive)

Crowdsourcing workers (non-expert, cheap)



Classifier trained using **expert** and **non-expert** labels achieved **AUC 0.962**

AUC: Area Under the receiver operator Curve

Background: Quality control of tasks posted to crowdsourcing has not received much attention

There are no guarantees for either worker or requester reliability → **quality control** is an important issue

Quality control of ***submissions*** has been well studied

Quality control of ***tasks*** has not received much attention

Motivation: Improper tasks are observed in crowdsourcing marketplaces

Example of improper task

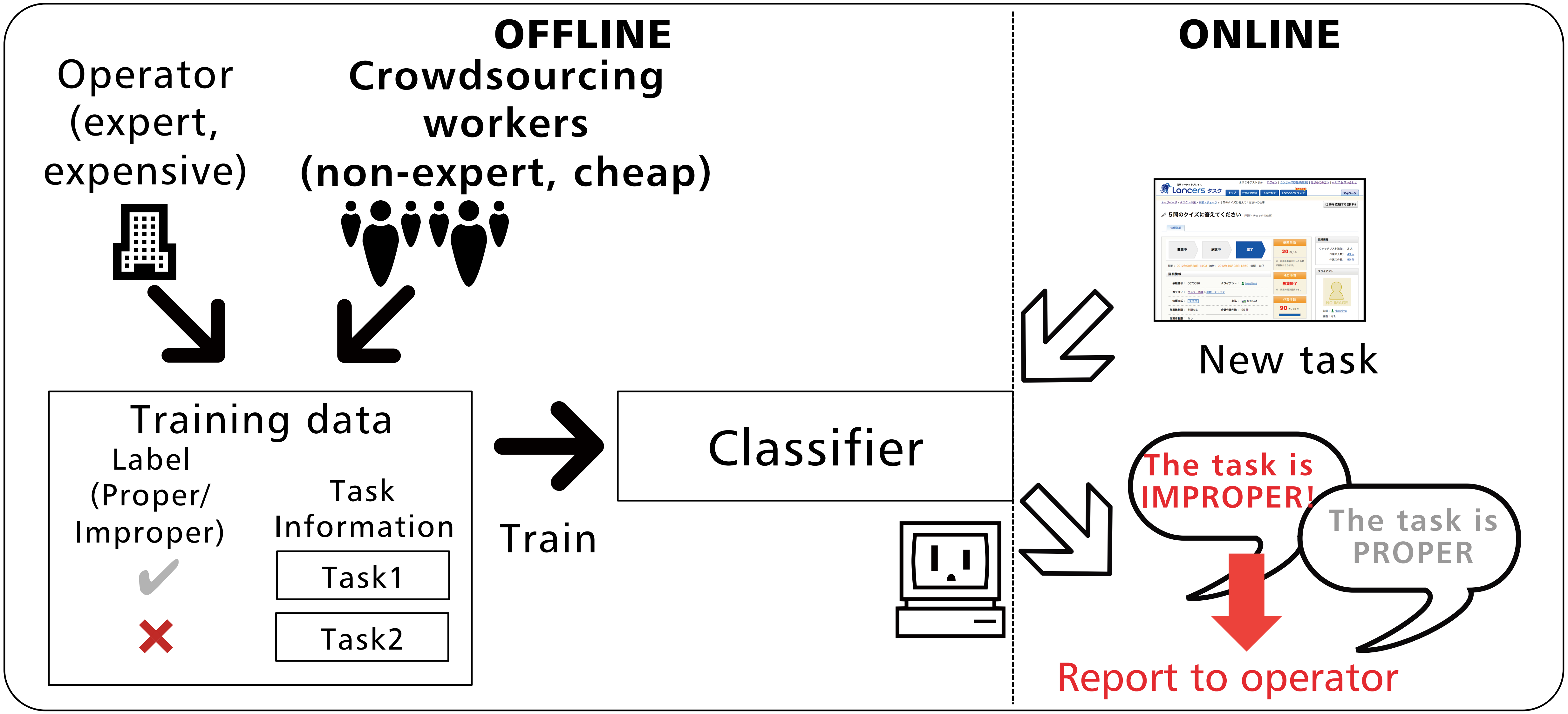
If you know anyone who might be involved in welfare fraud, please inform us about the person	
Name	
Address	

Other examples are

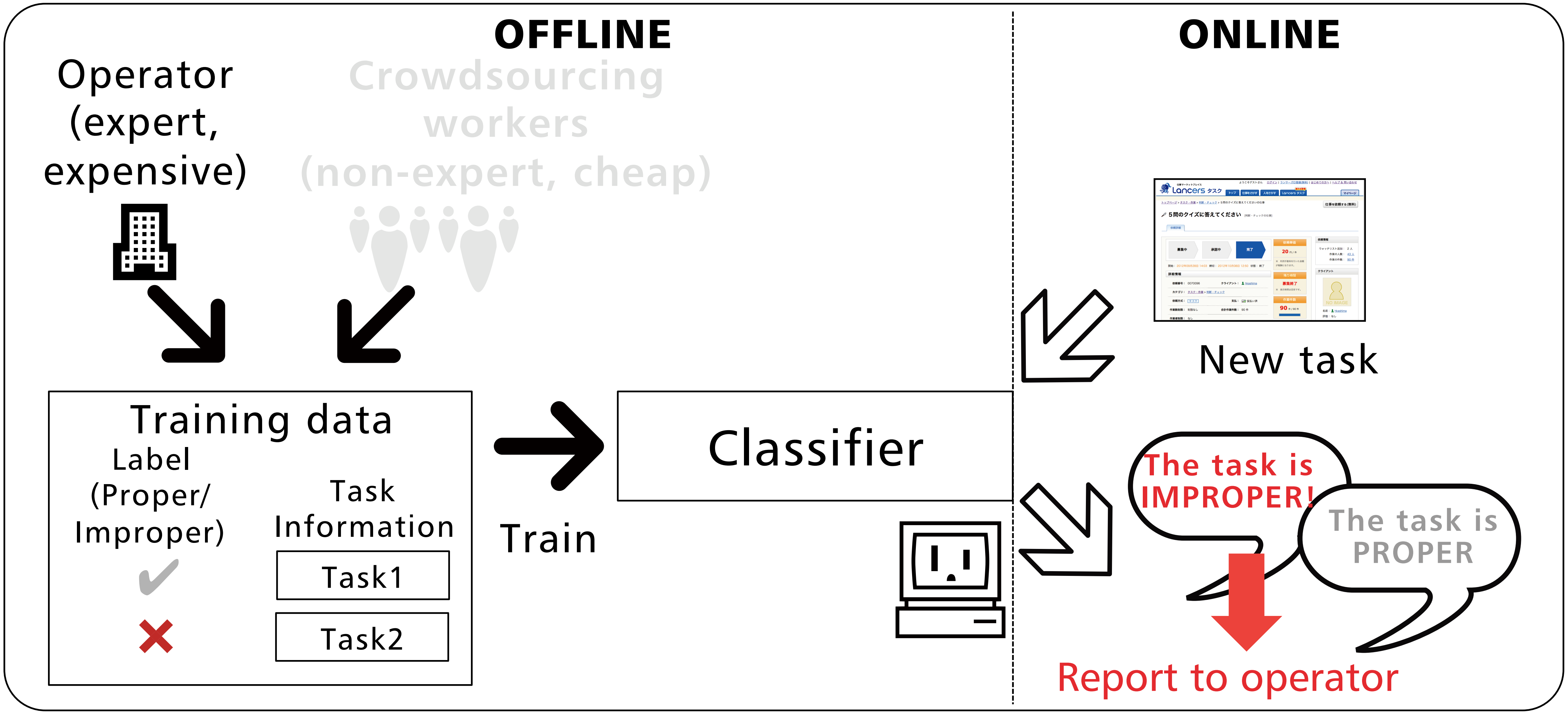
- Collecting personal information
- Requiring workers to register for a particular service
- Asking workers to create fake SNS postings

Operators in crowdsourcing marketplaces have to monitor the tasks continuously to find improper tasks. However, **manual investigation of tasks is very expensive**.

Goal: Supporting the manual monitoring by automatic detection of improper tasks



Experiment 1: We trained a classifier using labels given by expert operators



RESULTS

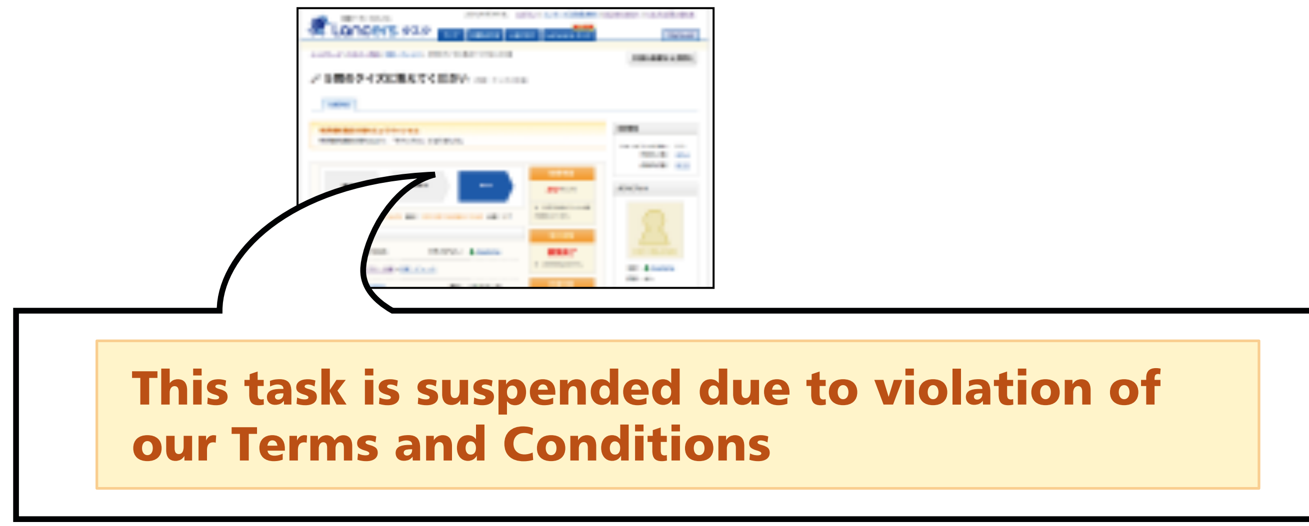
1. Classifier trained using expert labels achieved AUC 0.950
2. Classifier trained using expert and non-expert labels achieved AUC 0.962

Task dataset: Real operational data inside a commercial crowdsourcing marketplace

We used task data in クラウドソーシング Lancers
(MTurk-like crowdsourcing marketplace in Japan)

✔ 2,904 PROPER tasks

× 96 IMPROPER tasks



These labels are given by expert operators

Features used for training

Type	Examples or description
Task textual feature	Bag-of-words in task title and instruction
Task non-textual feature	Amount of reward and #assigned workers
Requester ID	“Who posted the task?”
Requester non-textual feature	Gender, age and reputation

Result 1: Classifier trained using expert labels achieved AUC 0.950

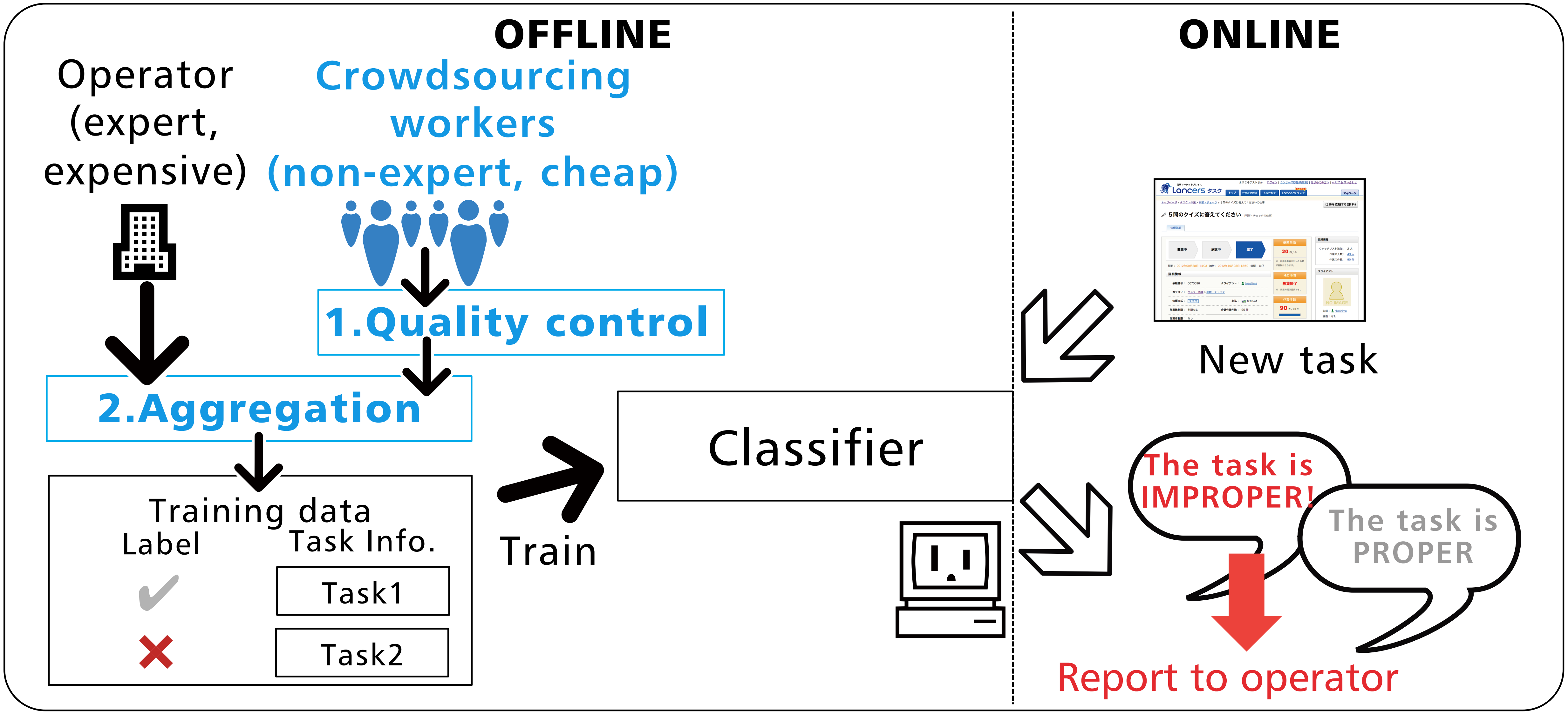
CLASSIFIER

Linear SVM, using 60% of tasks for training

RESULTS

- Classifier using all features showed the best performance (**0.950** for averaged AUC over 100 iterations)
- **The task textual feature** was the most effective
- Helpful features:
  - Red-flag keywords
    e.g., “account,” “password,” and “e-mail”
  - Amount of rewards
  - Requester reputation
  - Worker qualifications

Experiment 2: We trained a classifier using labels given by experts and non-experts



RESULTS

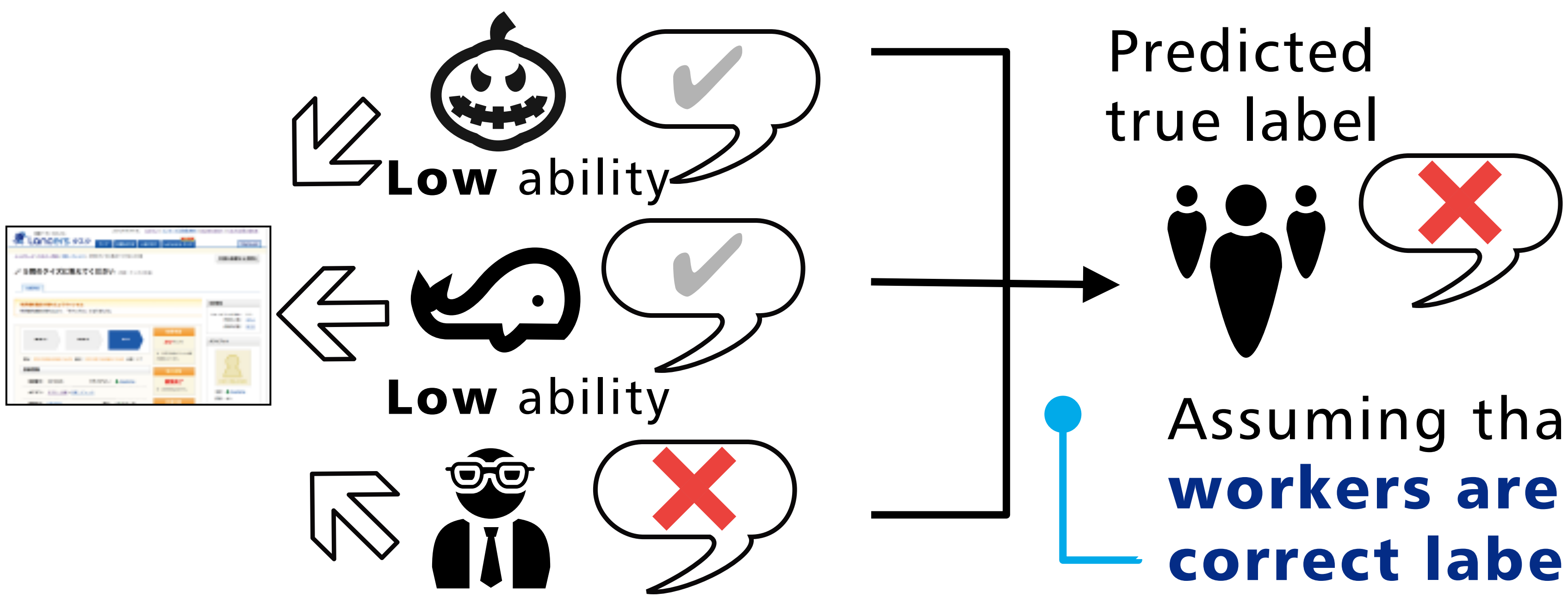
1. Classifier trained using expert labels achieved AUC 0.950
2. Classifier trained using expert and non-expert labels achieved AUC 0.962

Non-expert label dataset: Multiple workers were assigned to label each task

We hired crowdsourcing workers on クラウドソーシング Lancers and asked them to label each task

Quality control of worker labels: We applied existing method considering worker ability

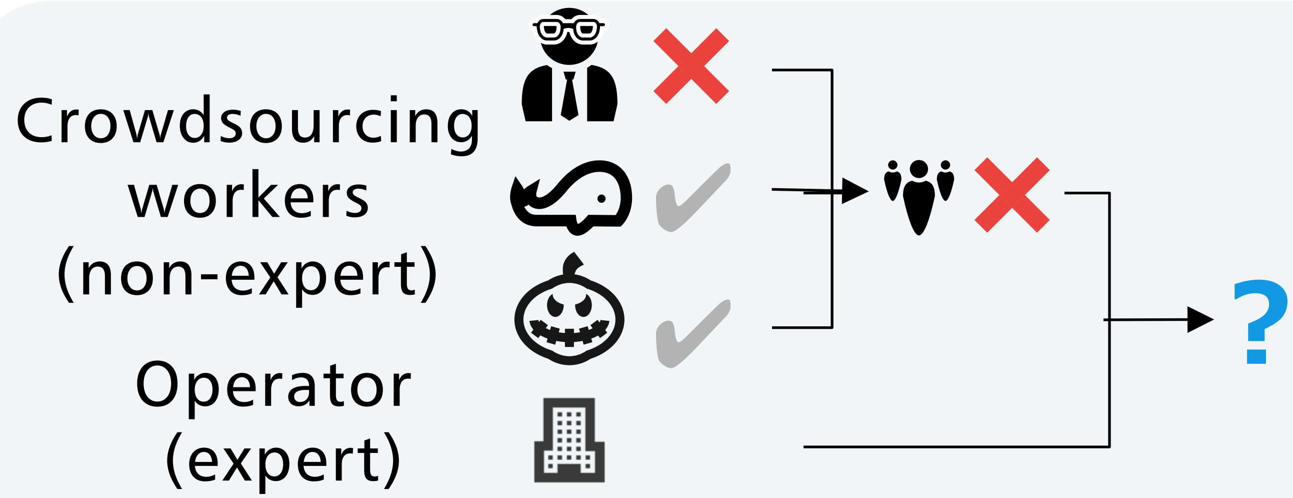
Reliability of labels given by non-expert workers depends on individual workers
→We merged the labels by applying a statistical method considering worker ability [Dawid&Skene ’79]



Assuming that “**High-ability workers are prone to give correct labels**” and “**Workers giving correct labels have a high ability**”: the method predicts both true labels and worker abilities

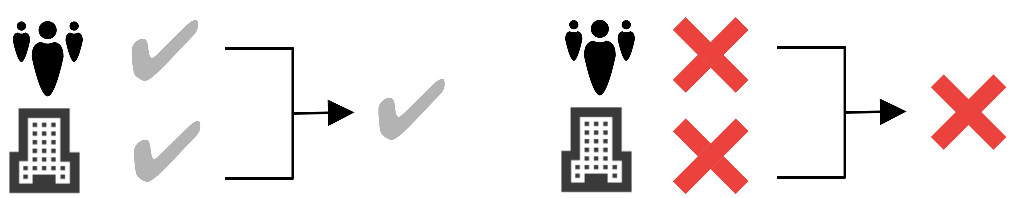
[Dawid&Skene ’79] A. P. Dawid and A. M. Skene.: Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm, Journal of the Royal Statistical Society, Series C (Applied Statistics) , Vol. 28, No. 1 (1979)

Aggregation of expert and non-expert labels: Rule-based strategies

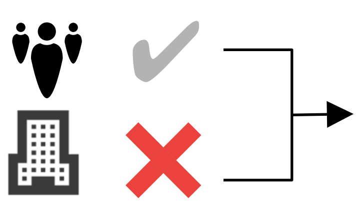


How can we aggregate labels given by experts and non-experts?

(1) If both labels are the same: just use the label



(2) If both labels are different: We have three strategies

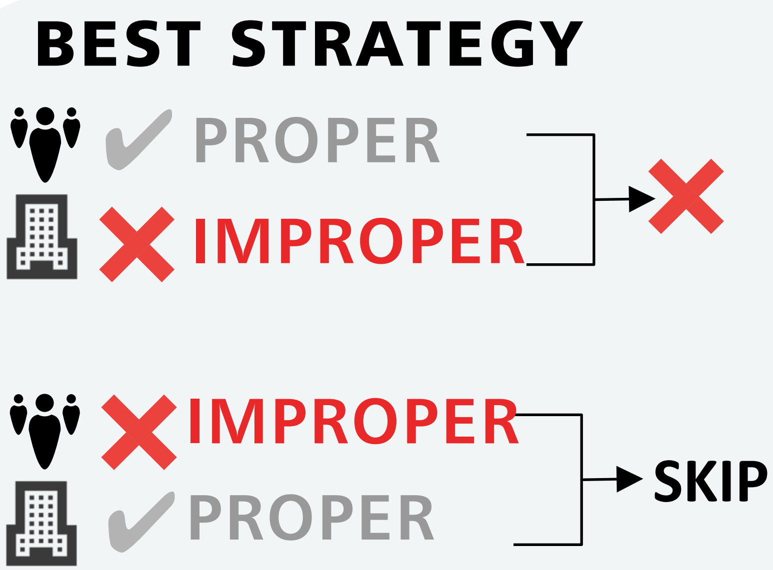


1. Select ✔ PROPER as agreed label
2. Select ✖ **IMPROPER** as agreed label
3. Ignore the sample (labeled task) and do not include it in training dataset

We have 3×3 strategies for both cases in ✔ ✖ and ✖ ✔

Result 2: Classifier trained using expert and non-expert labels achieved AUC 0.962

Best strategy achieved averaged **AUC 0.962**



If the experts judge a task as **IMPROPER**, the strategy always sides with the experts.

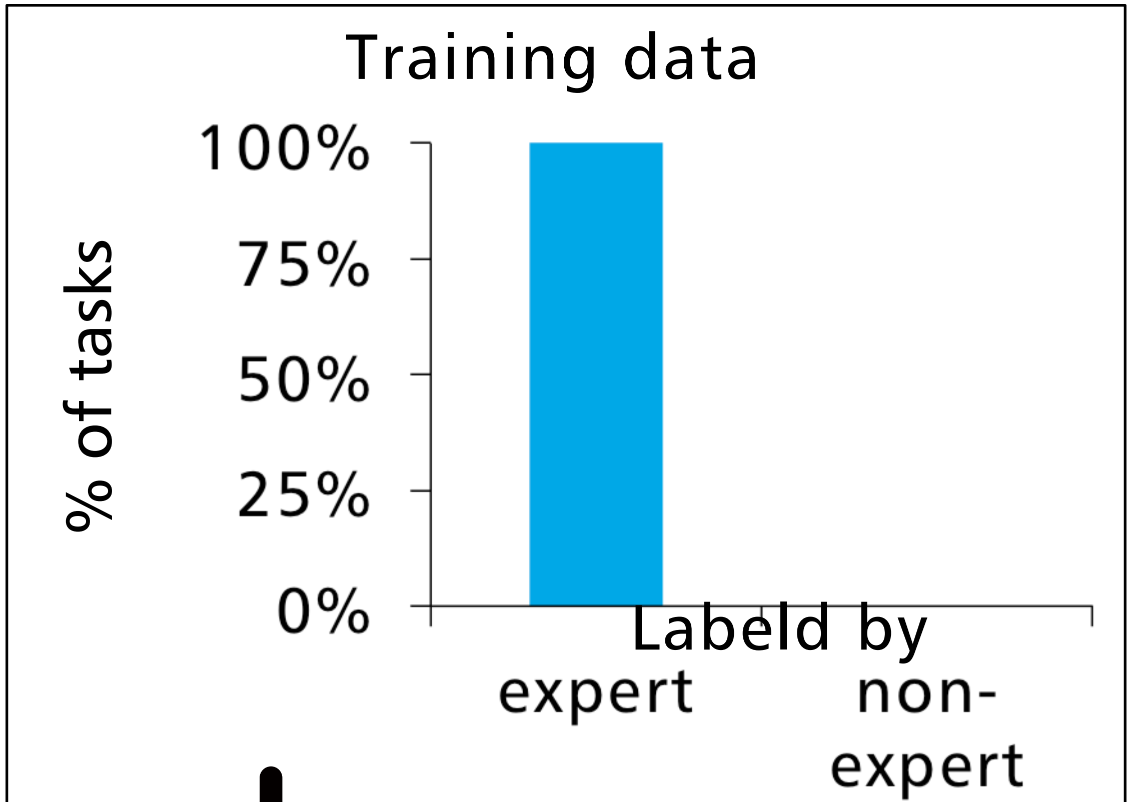
If the experts judge a task as **PROPER** but the non-experts disagree, the strategy ignores the sample.

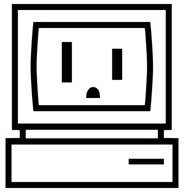
Why did the best strategy perform well?

The expert operators seem to judge a task as IMPROPER strictly, so we should sides with them if they judge a task as IMPROPER.

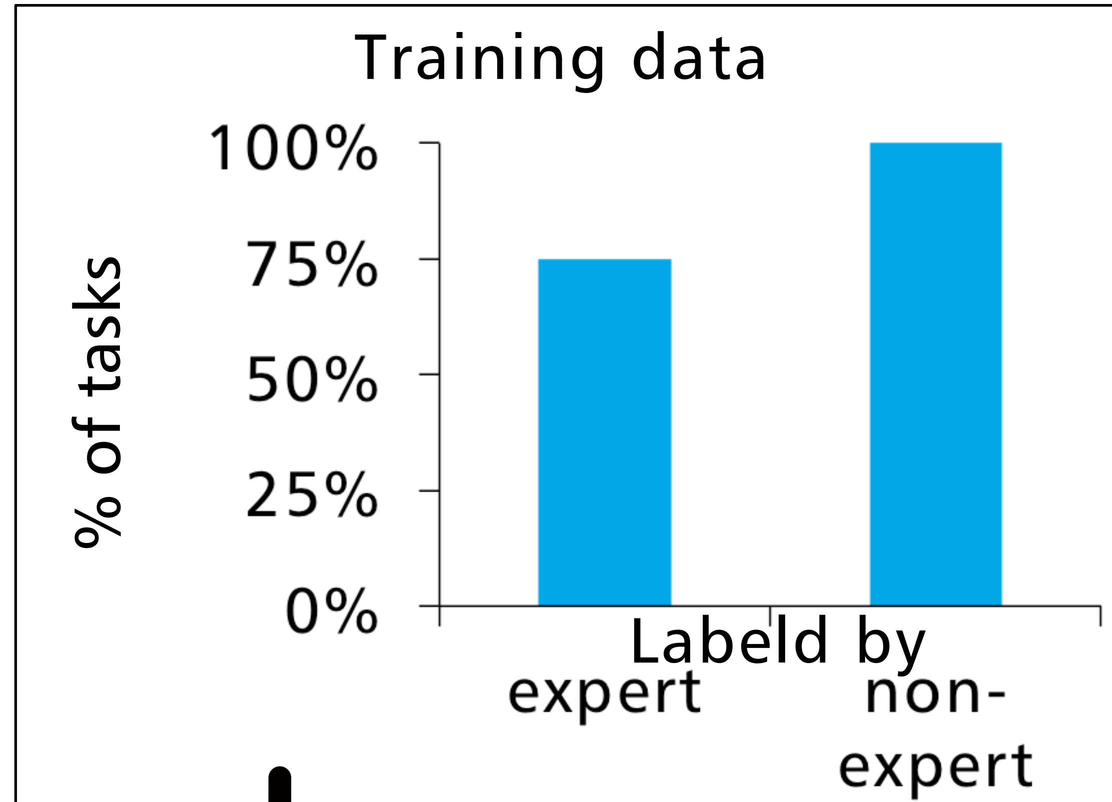
The non-expert crowdsourcing workers are not very serious to judge a task as IMPROPER so we should ignore the task if the experts disagree the judgments by the non-experts.

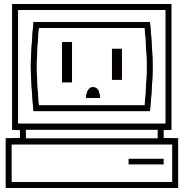
Result 3: Crowd labels can reduce the number of expert labels by 25% while maintaining accuracy





AUC 0.950
(Result 1)





AUC 0.951
(Result 3)

Crowdsourced labels are useful in reducing the number of expert labels while maintaining the same level of classification performance

Summary: Crowdsourced labels are useful for detecting improper tasks in crowdsourcing

To support manual monitoring, we used machine learning techniques and built classifiers to detect improper tasks in crowdsourcing

1. ML approach is effective in improper task detection (Result 1)
2. By addressing a range of reliability and choosing a good aggregation strategy, crowdsourced labels improved the performance of improper task detection (Result 2)
3. Crowdsourced labels are useful in reducing the labeling cost of expert operators (Result 3)