Understanding Large-Scale I/O Workload Characteristics via Deep Neural Networks

Jinyoung Moon and Myoungsoo Jung

Outline

1. Introduction and Background
2. Performance metrics
3. Modeling and Training
4. Challenge #1: Low Balanced Accuracy
5. Challenge #2: Consistency of Performance
6. Result

SSD Background Tasks

Intel SSD 730 240GB

Garbage Collection (GC)

Read Reclaim

Wear leveling

Page Migration

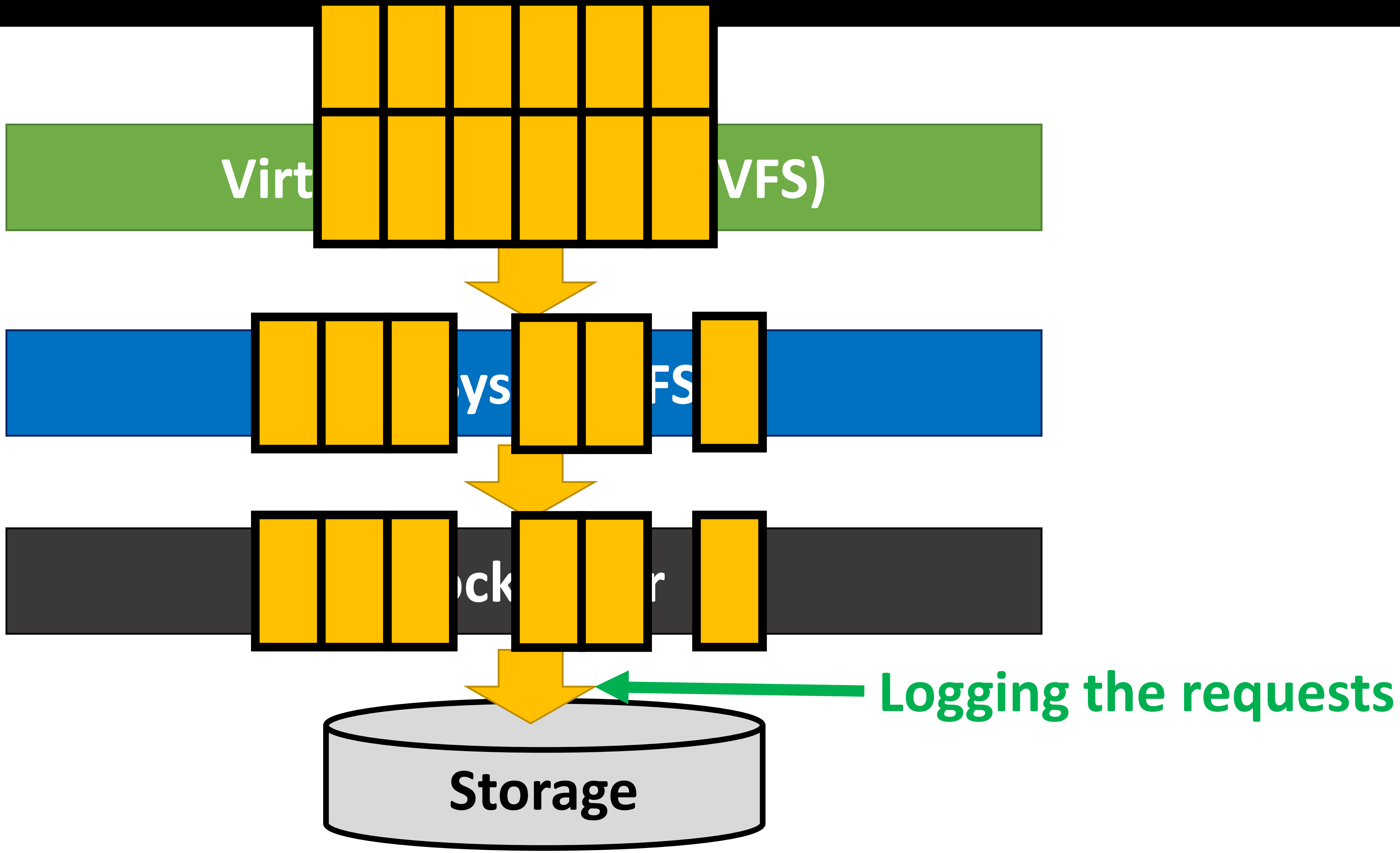
- SSD performs background tasks that **consume much time** than Read/Write operation.
- **Improve SSD performance:** Performing background tasks in **idle time**.

http://www.bit-tech.net/

Long Time Interval

- The long time interval is expected to have an idle time.
- Perform background tasks during idle time.
  - **When** will the next request come? *Can we predict?*

I/O Workload



I/O Workload

```
0.195966428   Write   917576   8   Rand
0.205976081   Write   917600   8   Rand
0.215986100   Write   5283576 8   Rand
```

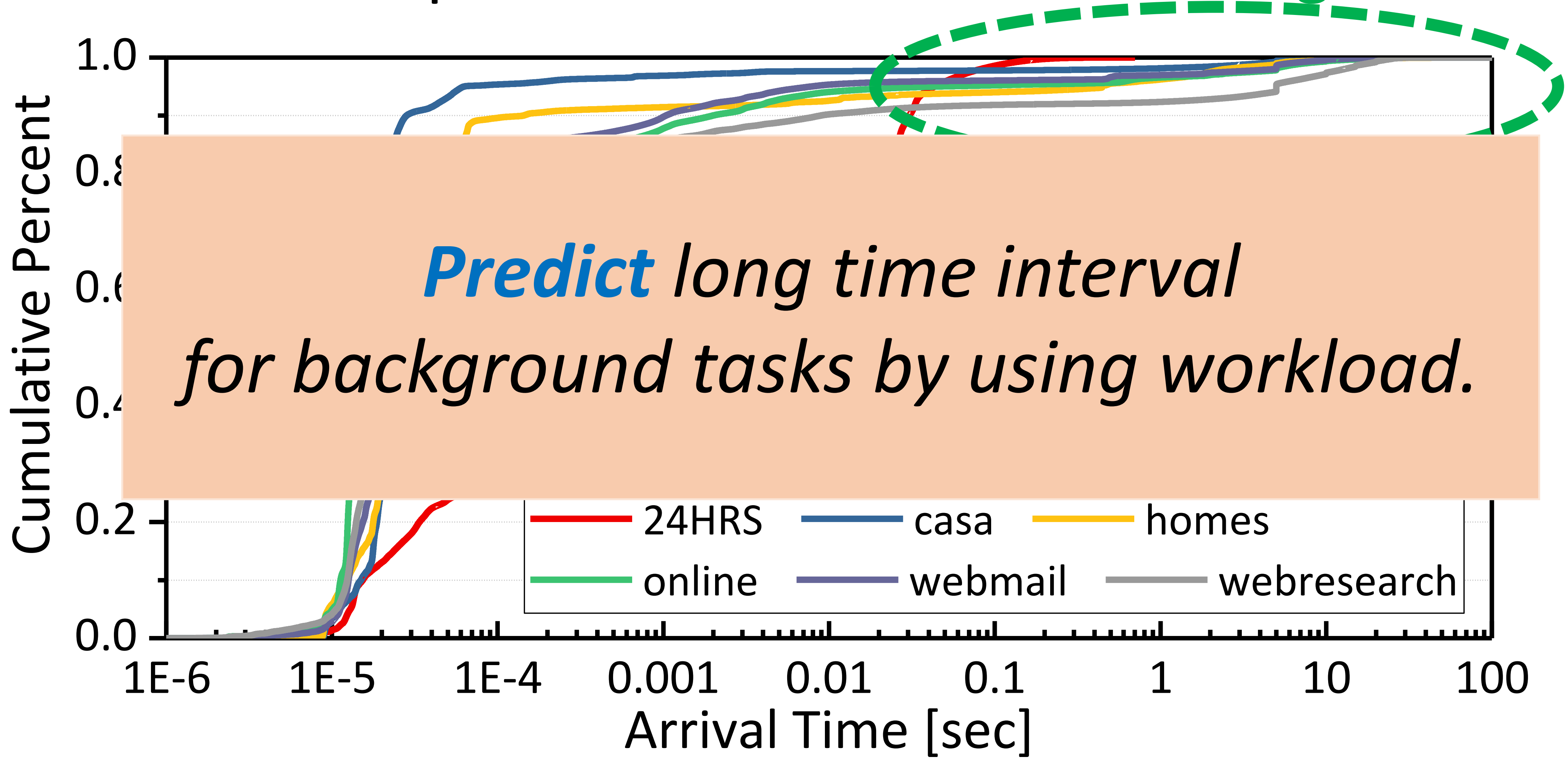
⋮

Time stamp, Operation Type, LBA, Sector Size, Access Type, ...

Characteristic, Locality, Access Pattern, ...

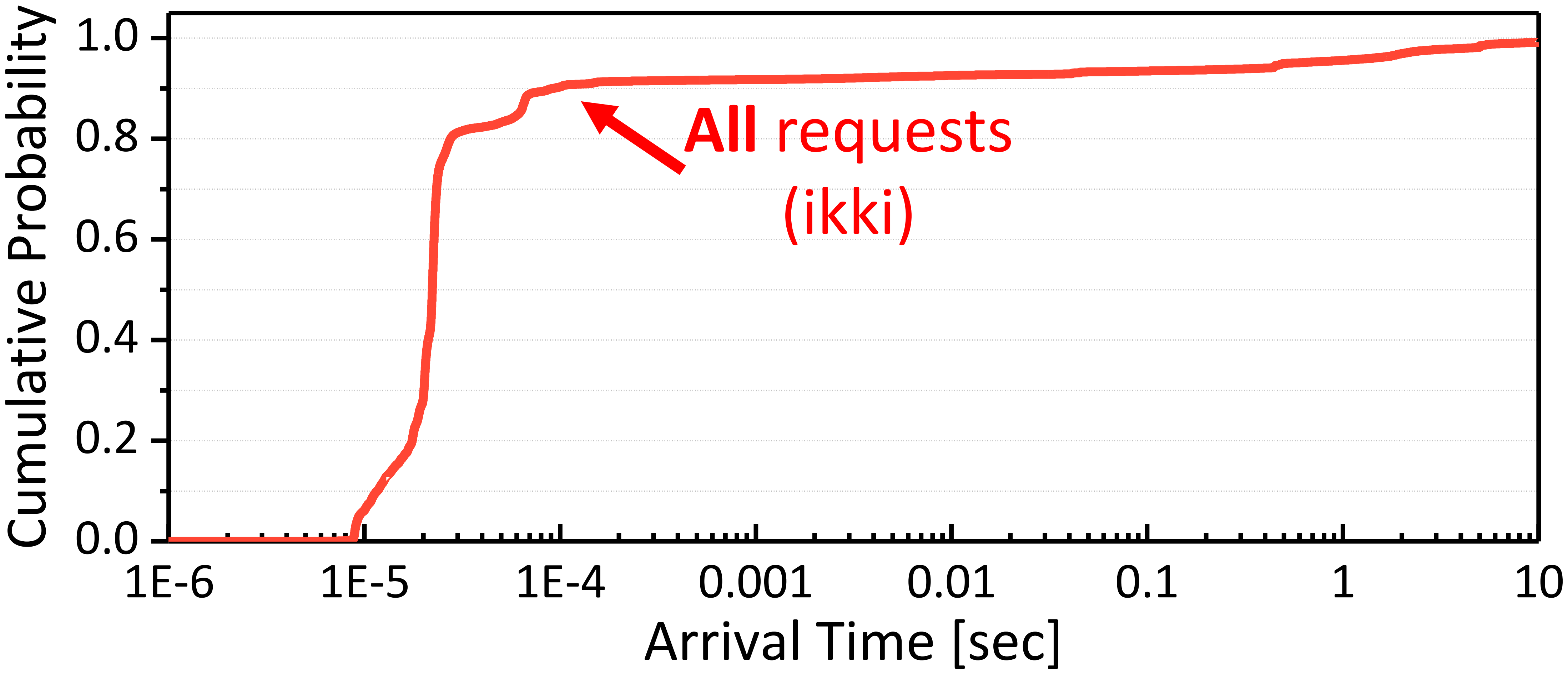
Long Time Interval

There are some requests which **arrive after a long time.**

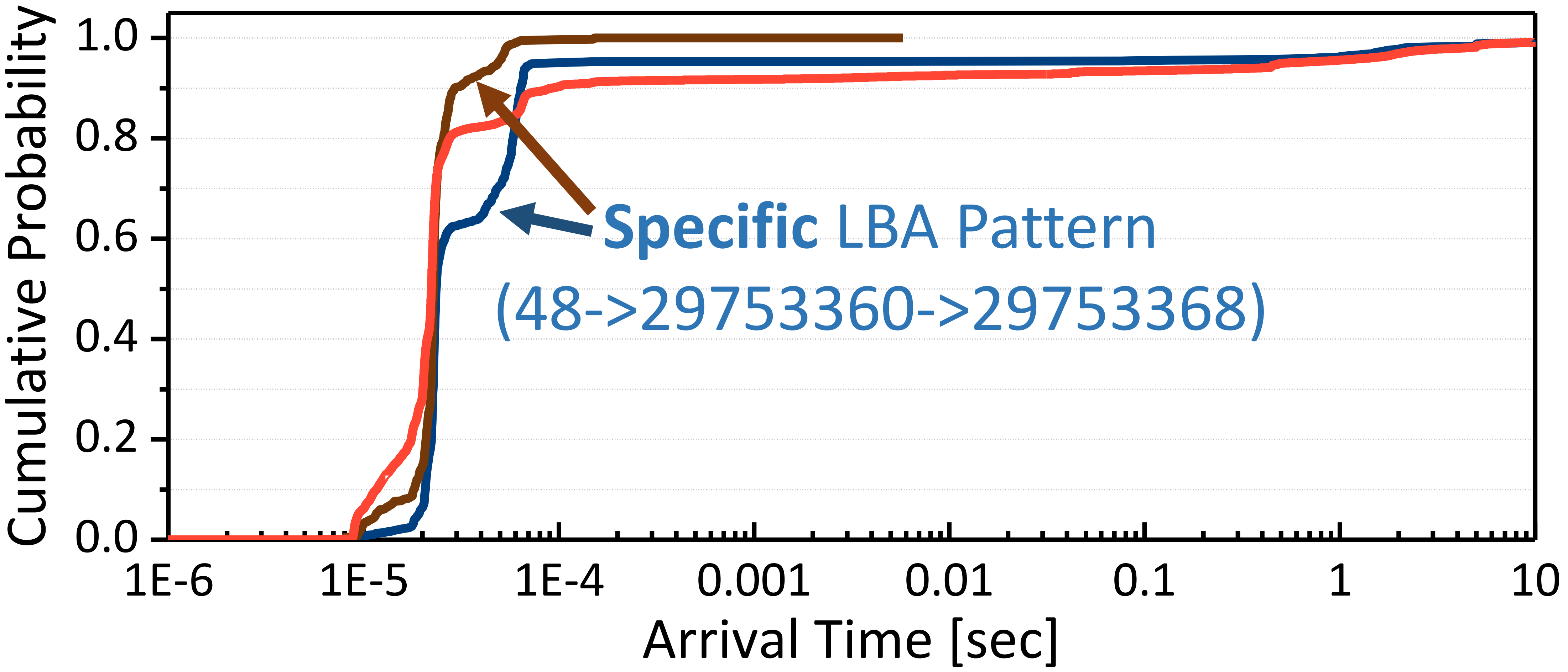


***Predict** long time interval*
for background tasks by using workload.

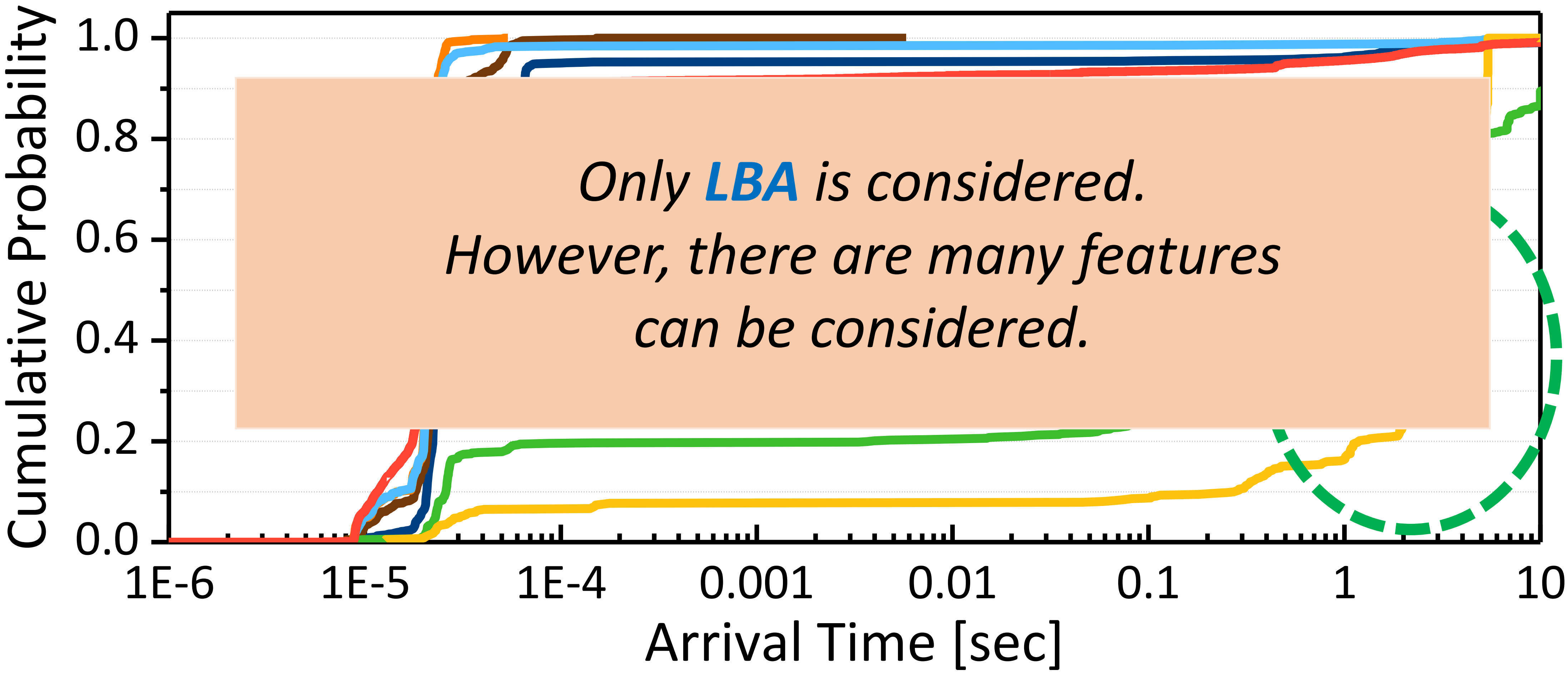
Arrival Time of Address Patterns



Arrival Time of Address Patterns



Arrival Time of Address Patterns



*Only **LBA** is considered.*
However, there are many features
can be considered.

Arrival Time of Address Patterns

*It is **difficult** to **find a pattern** considering several features.*

- Several features mean **higher dimensions**.
- Higher dimensions not only make works **harder**, but also require **a lot of time**.

*Each workloads has **different characteristics**.*

- Patterns can be found in **different rules**.

*Using a **Deep Neural Network** models!*

Outline

1. Introduction and Background

2. Performance metrics

3. Modeling and Training

4. Challenge #1: Low Balanced Accuracy

5. Challenge #2: Consistency of Performance

6. Result

Accuracy

- **Accuracy**

: How well the model fit in the overall data.

: $$\frac{\sum_{i=0}^{N} a_{ii}}{\sum_{i=0}^{N}\sum_{j=0}^{N} a_{ij}} = \frac{95+30}{95+5+20+30} = \mathbf{0.833}$$

	Predict 0	Predict 1
Target 0	95 a_{00}	5 a_{01}
Target 1	20 a_{10}	30 a_{11}

Confusion Matrix

Target 1 Accuracy: 0.6
(Low)

*It is **not sufficient** to represent the performance!*

Balanced Accuracy

- **Balanced Accuracy**

: The accuracy of each classes.

: Target 0:   0.95

: Target 1:   0.60 (Low)

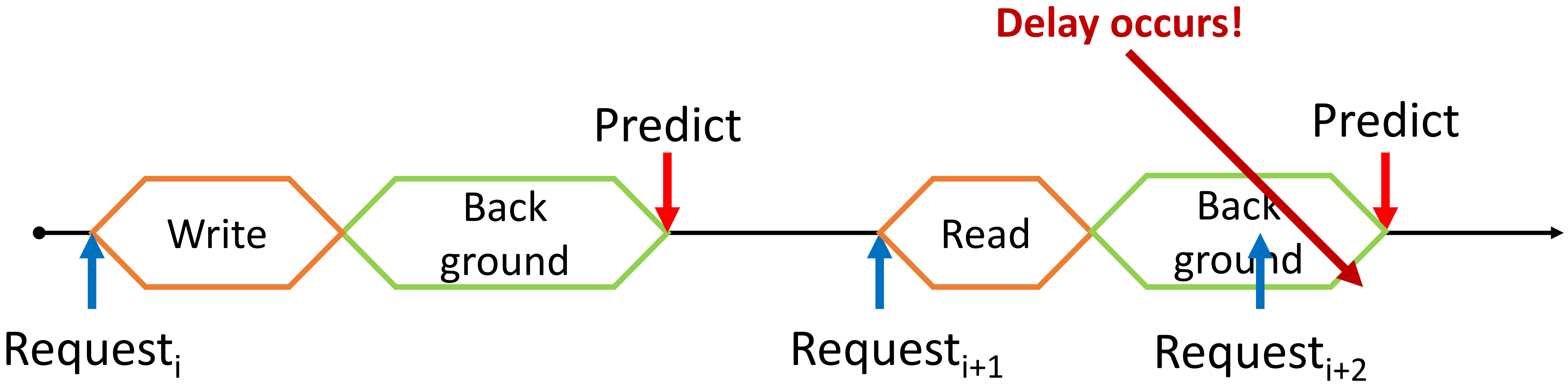
- **Average of Balanced Accuracy**

: $\frac{1}{N}\sum_{i=0}^{N}\frac{a_{ii}}{\sum_{j=0}^{N}a_{ij}} = \frac{1}{2}\cdot\left(\frac{95}{95+5}+\frac{30}{20+30}\right) = \mathbf{0.775}$

	Predict 0	Predict 1
Target 0	95 a_{00}	5 a_{01}
Target 1	20 a_{10}	30 a_{11}

Confusion Matrix

Penalty



Delays due to mispredictions **degrade** performance.

- **Reduce the probability of penalty.**
- Early stop background tasks when a penalty occurs.

Non-Penalty

- **Non-Penalty**

: The rate when delays didn't occur.

$$: 1 - \frac{\sum_{i=0}^{N-1}\sum_{j=i+1}^{N} a_{ij}}{\sum_{i=0}^{N}\sum_{j=0}^{N} a_{ij}} = 1 - \frac{5}{95+5+20+30}$$

$$= \mathbf{0.967}$$

	Predict 0	Predict 1
Target 0	95 a_{00}	5 a_{01}
Target 1	20 a_{10}	30 a_{11}

Confusion Matrix

<u>Goal:</u> **high** average of balanced accuracy and **low** penalty

Outline

1. Introduction and Background

2. Performance metrics

3. Modeling and Training

4. Challenge #1: Unbalanced data

5. Challenge #2: Consistency of Performance

6. Result

Training

- Input features
  : LBA, Operation type, Access type, Sector Size
- Output Class
  : Class 0, Class 1 (Threshold: 0.1 sec)

Goals:

1. Prevent overfitting and underfitting.
2. Make **high** avg. balanced accuracy and **low** penalty.

Using training techniques and tuning hyper parameters.

Input Data and Target Data

Target data 1 / **Target data 2**

0.195966428	Write	917576	8	Rand
0.205976081	Write	917600	8	Rand
0.215986100	Write	5283576	8	Rand
⋮				
0.226338105	Write	5284578	8	Rand
0.747671305	Write	1255342	8	Rand
0.751198905	Write	1255350	8	Seq
0.753334136	Write	1255358	8	Seq

Time-series Information

Input data 1 / **Input data 2**

N steps

Related to each other

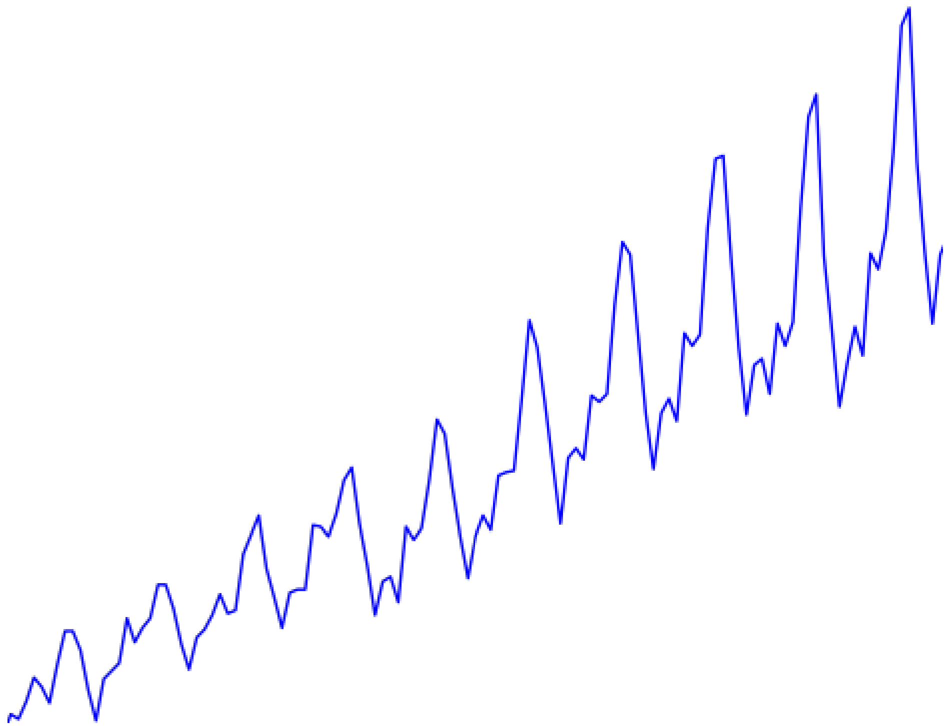
0.521 > 0.1 sec

: Class 1 / **0.003 < 0.1 sec**

: Class 2 / **0.002 < 0.1 sec**

: Class 0

Recurrent Neural Network (RNN)



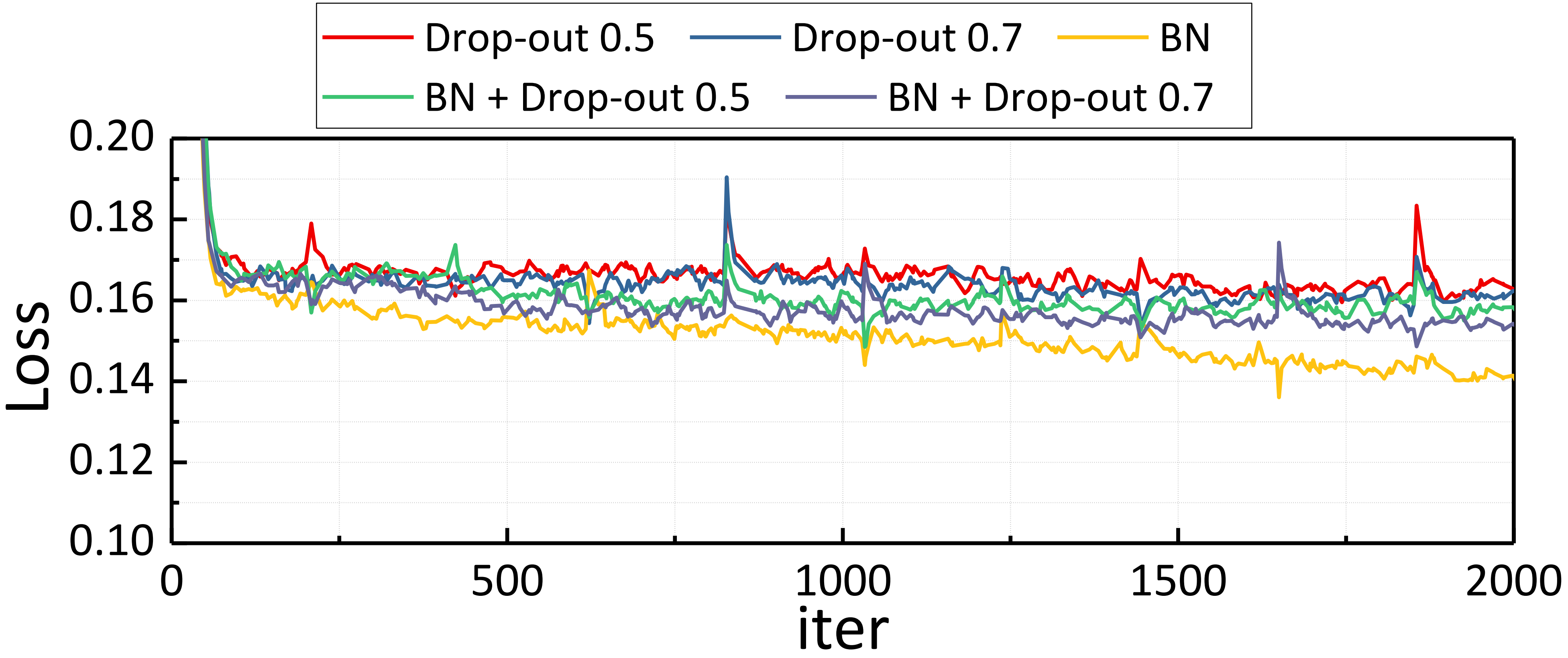
… **Next ?**

*RNN is used to train a **time-series data**.*
Ex) stock forecast, translation, …

Common Training Techniques

1. **Drop-out**
   : Learns a certain percentage of neurons.
2. **Early Stopping**
   : Stop learning before overfitting occurs. (Default)
3. **Batch Normalization**
   : Normalize, scaling and shift input data in front of hidden layer.
4. **Hyper parameter tuning**
   : Change learning rate, # neurons, # layers, step size, …

Loss by using Train Techniques



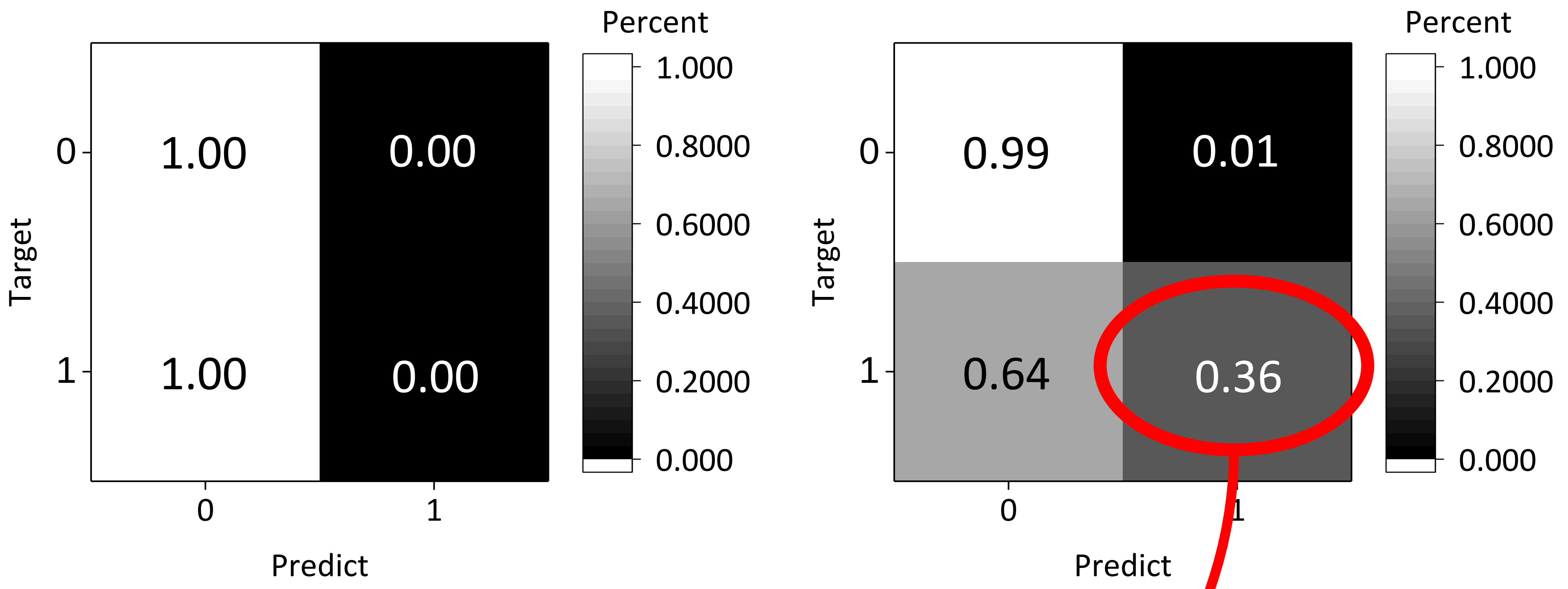
webmail

- The loss of **BN** (Batch Normalization) **decreases most rapidly**.
- Using only **BN** is expected to show the **best** performance.

Outline

1. Introduction and Background
2. Performance metrics
3. Modeling and Training
4. Challenge #1: Low Balanced Accuracy
5. Challenge #2: Consistency of Performance
6. Result

Challenge #1. Low Balanced Accuracy



(a) Drop-out

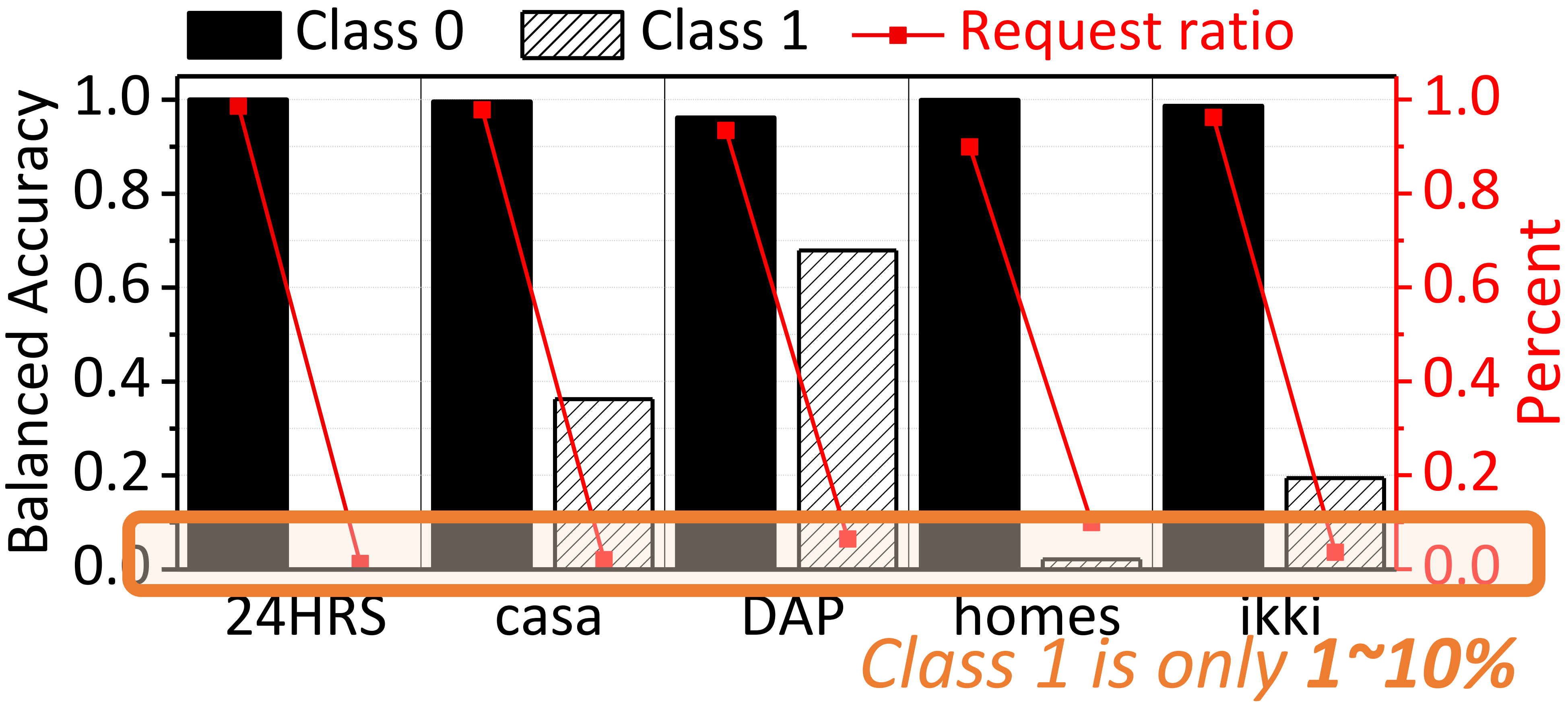
(b) Batch Normalization

casa

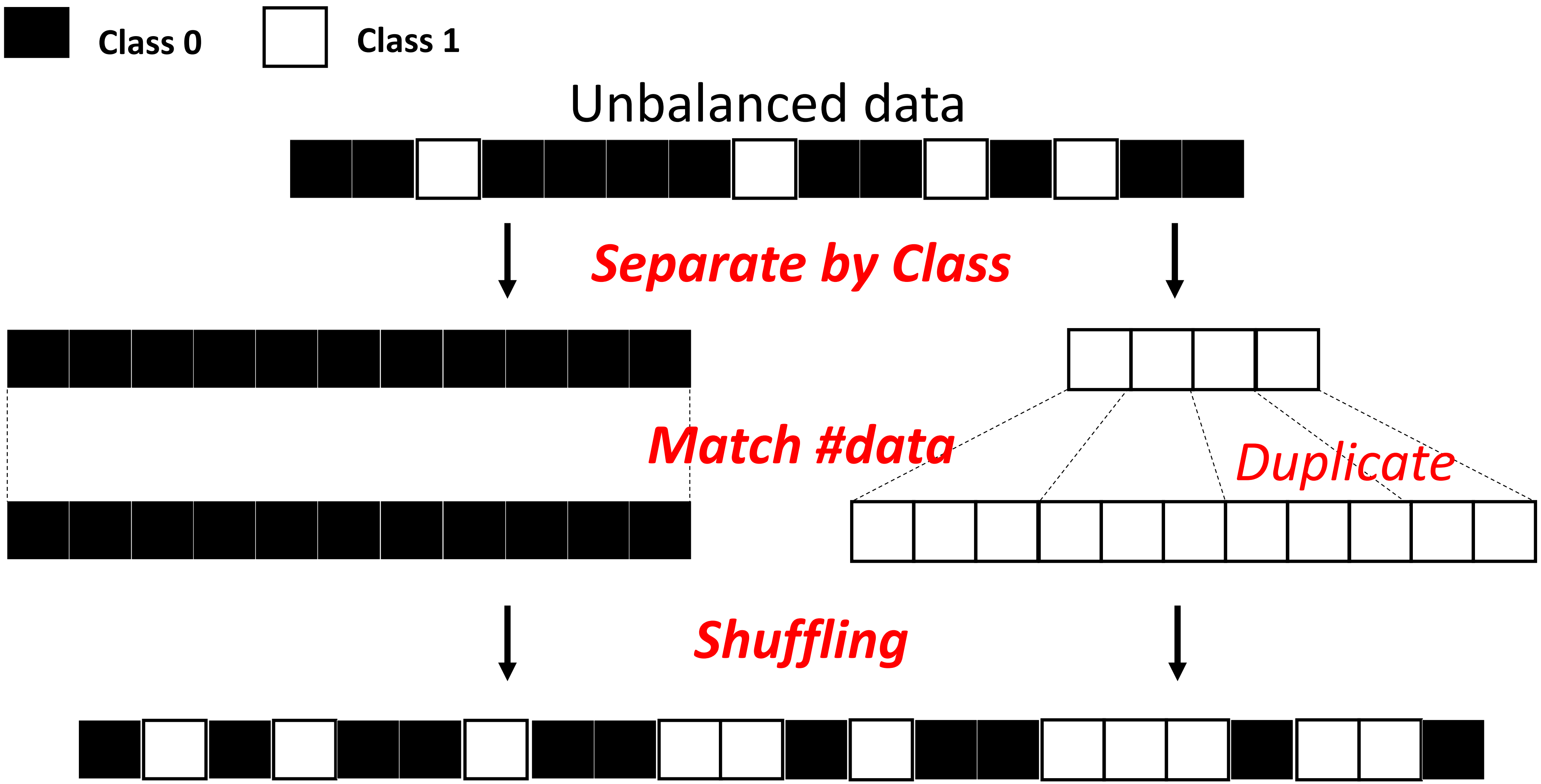
*Target 1 balanced accuracy is very **low**!*

Unbalanced of Class

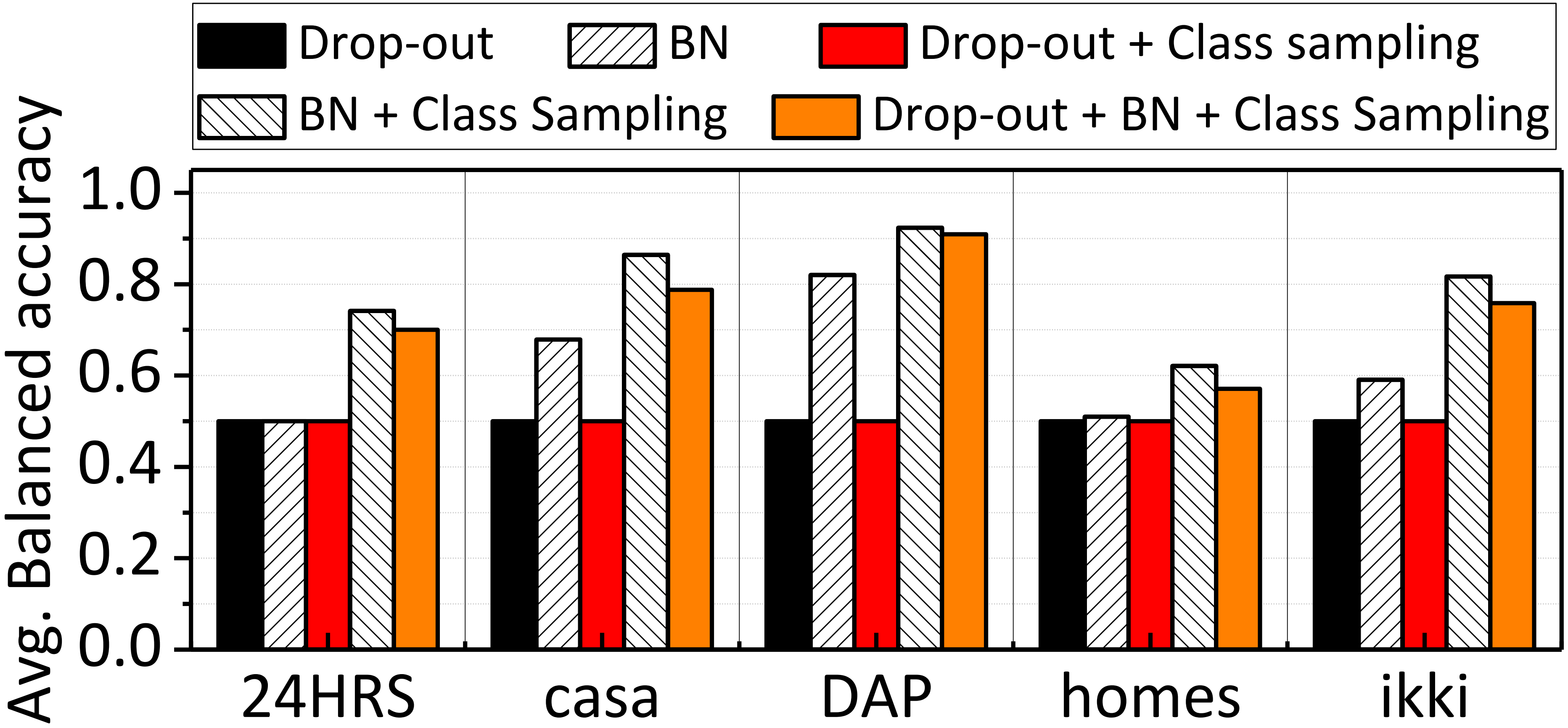
The unbalanced data makes **training difficult**.



Solution. Up-Sampling by Class

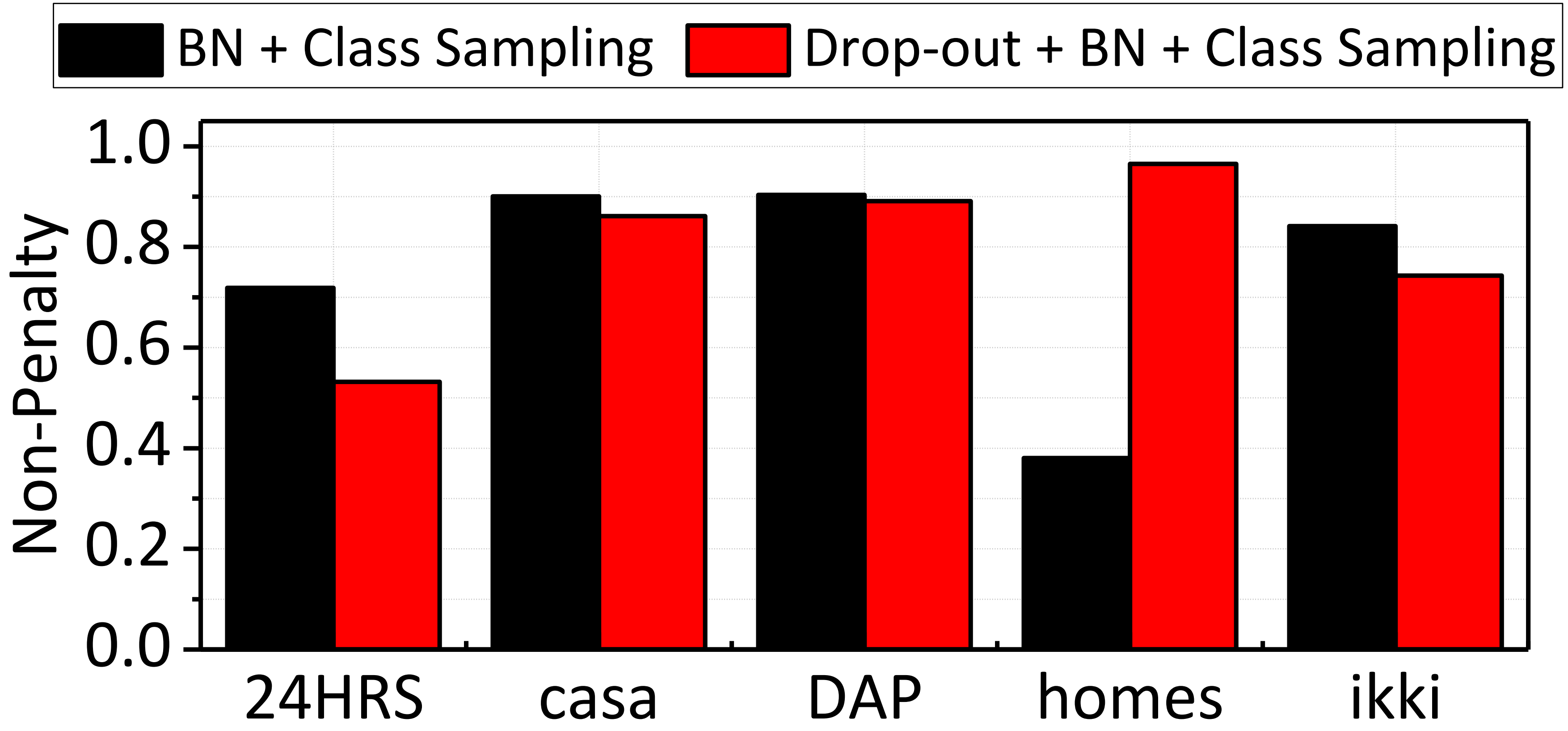


Compare each Training Techniques



In most cases, **BN+Class Sampling** and **Drop-out+BN+Class Sampling** show high performance.

Compare BN and Drop-out+BN



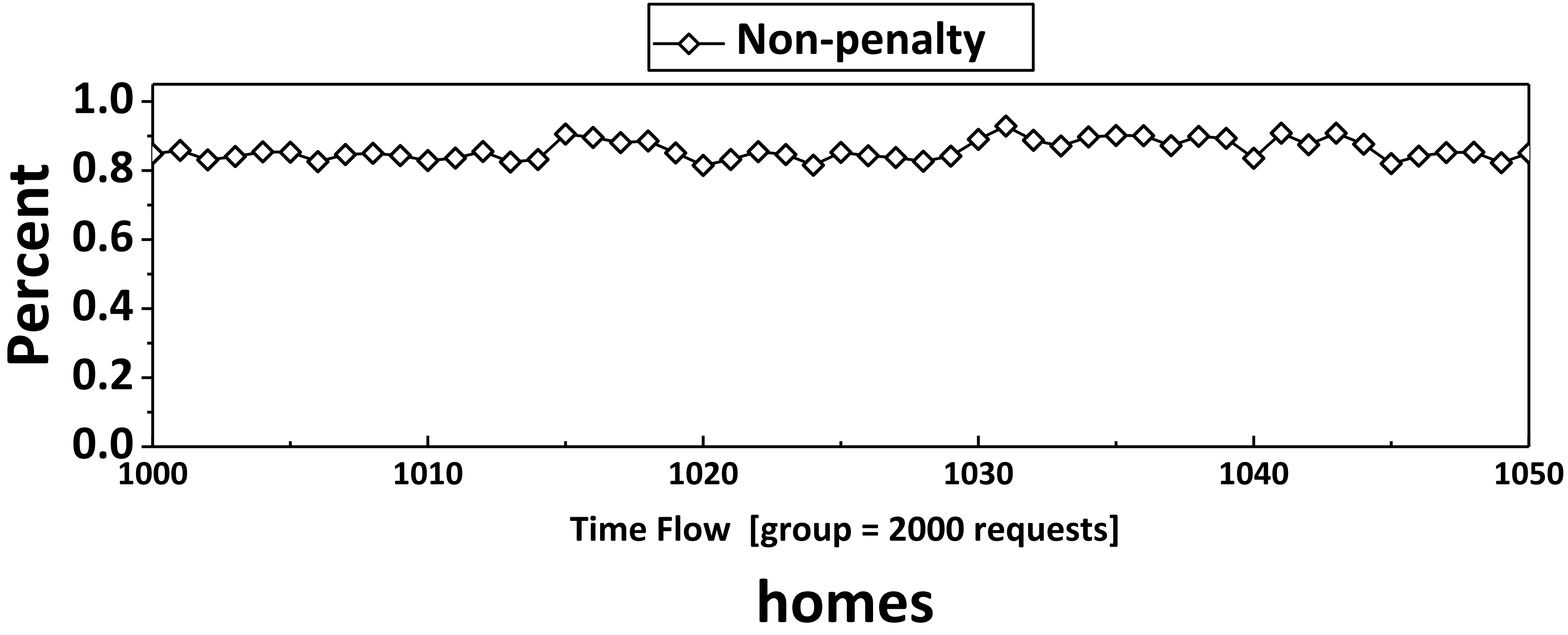
In most cases, **BN+Class Sampling** is better than using **Drop-out**.

Outline

1. Introduction and Background
2. Performance metrics
3. Modeling and Training
4. Challenge #1: Low Balanced Accuracy
5. Challenge #2: Consistency of Performance
6. Result

Challenge #2. Consistency of Performance

Is consistent of high non-penalty is maintained?

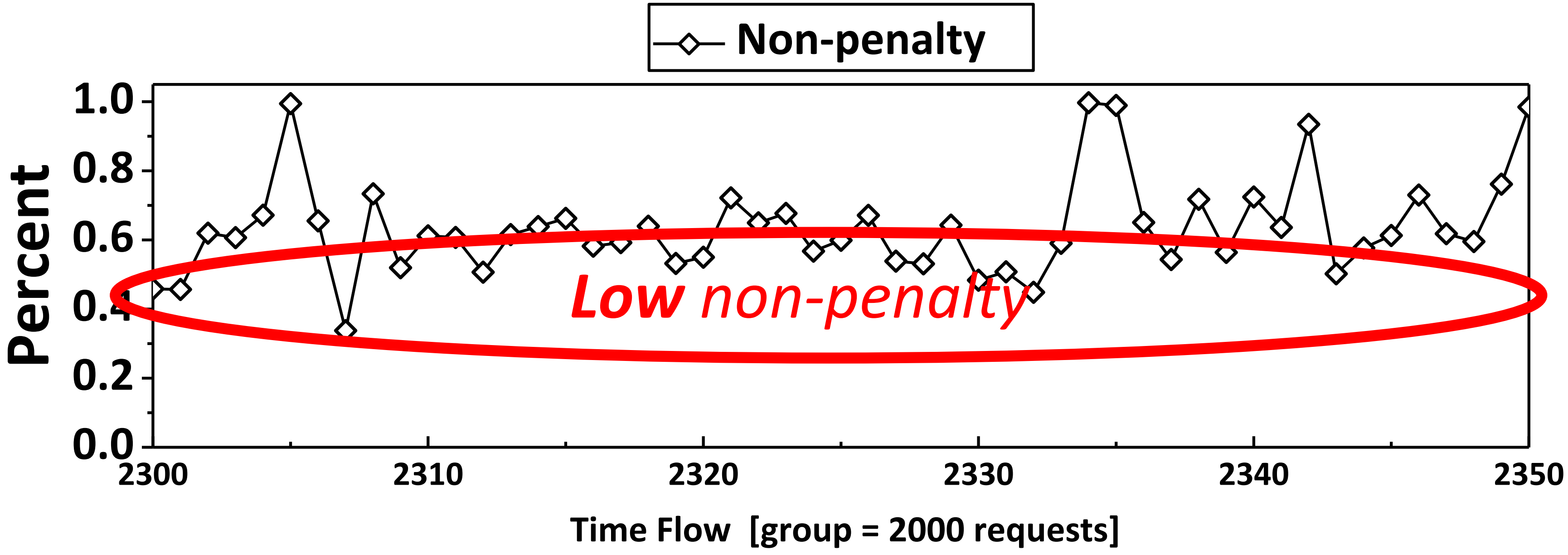


In this case, the **consistency** of performance looks like good.

Challenge #2. Consistency of Performance

Is consistent of high non-penalty is maintained? ***No!***

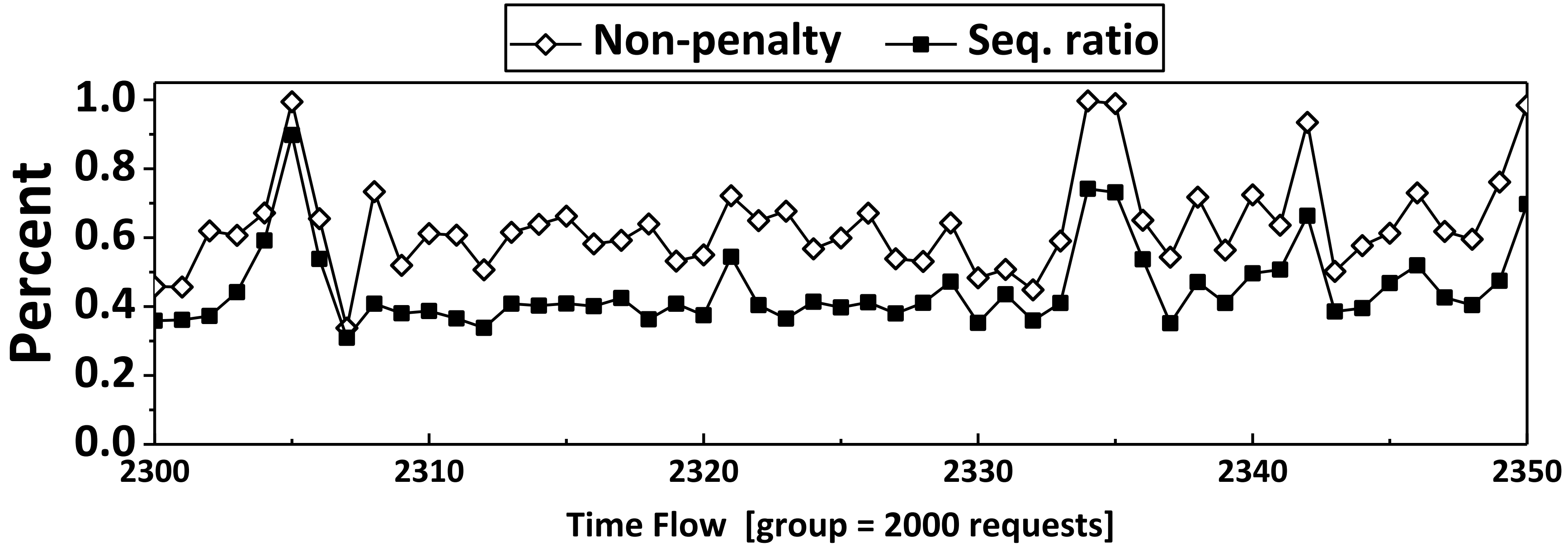
The **low** non-penalty interval can **adversely affect** on SSD.



homes

Challenge #2. Consistency of Performance

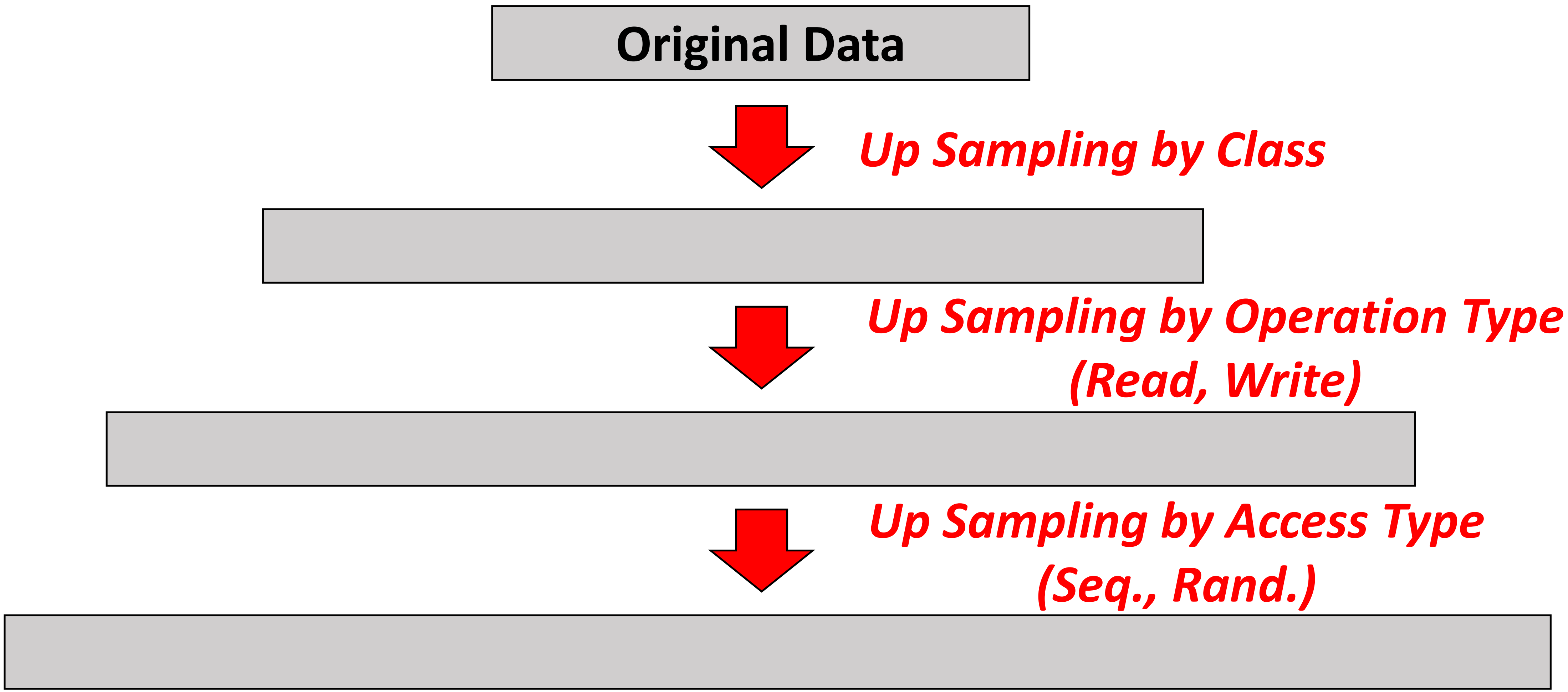
The tendency of non-penalty and seq. ratio is **similar**.



homes

The performance of predicting a random access is **bad**!

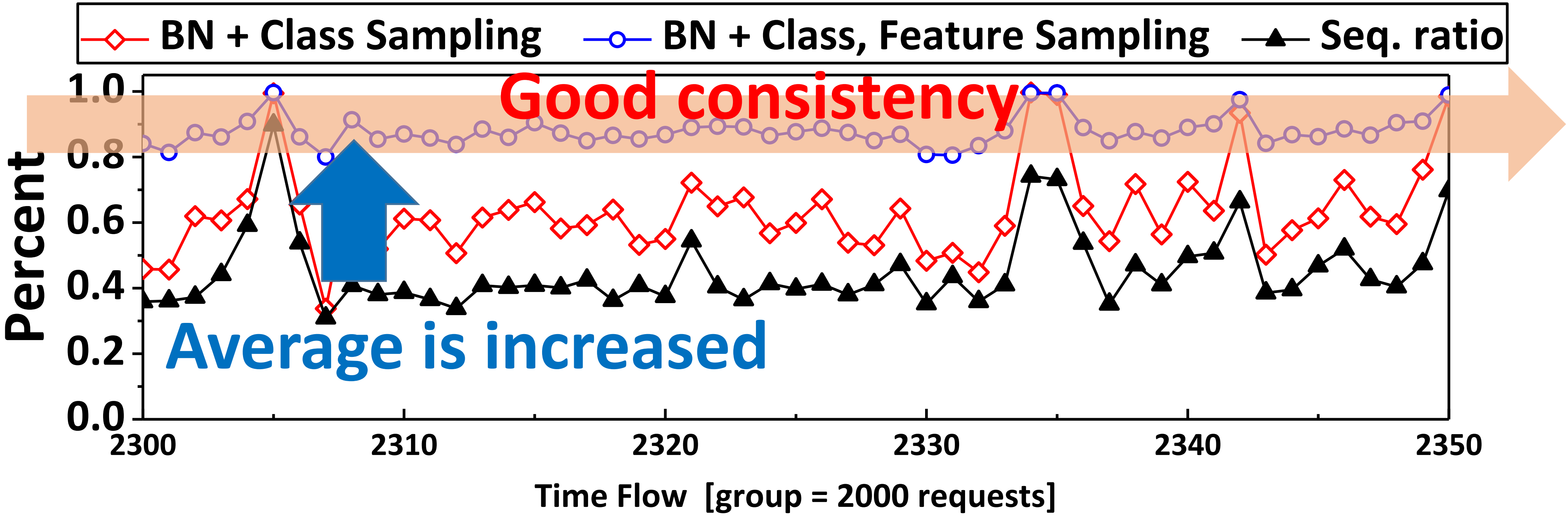
Solution. Up-Sampling by Features



Outline

1. Introduction and Background
2. Performance metrics
3. Modeling and Training
4. Challenge #1: Low Balanced Accuracy
5. Challenge #2: Consistency of Performance
6. Result

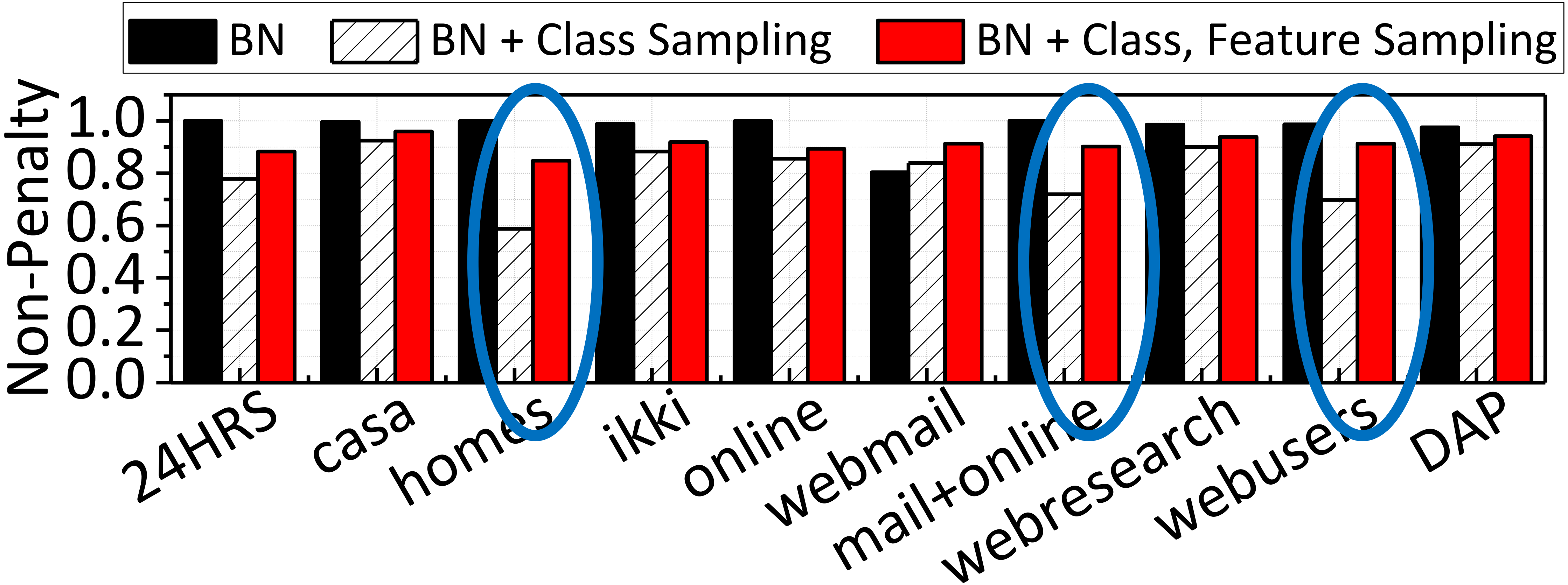
Result



homes

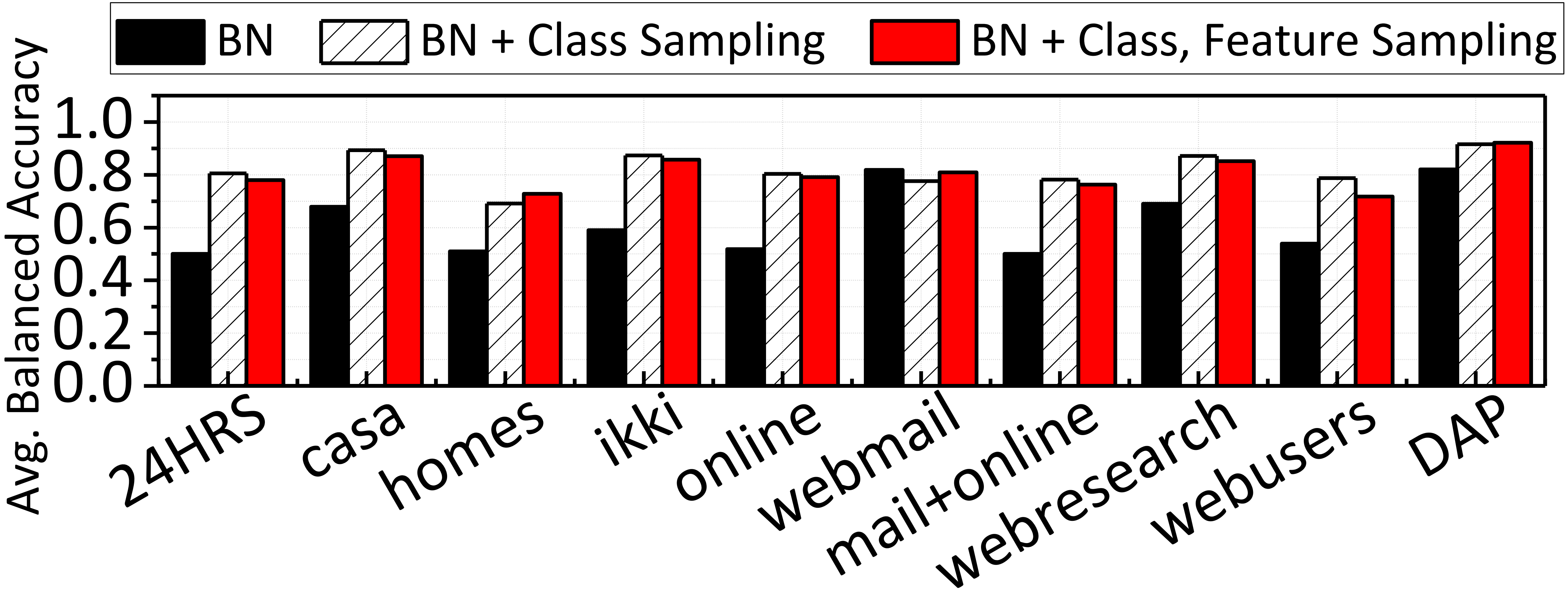
- Applying **Feature Sampling**, the **consistency** of non-penalty is improved.
- In addition, the average of non-penalty is **increased**.

Non-penalty



- **BN** occurs **underfitting**.
- Achieve **high** non-penalty and good **consistency**.
- The average of non-penalty is **90.79%**.

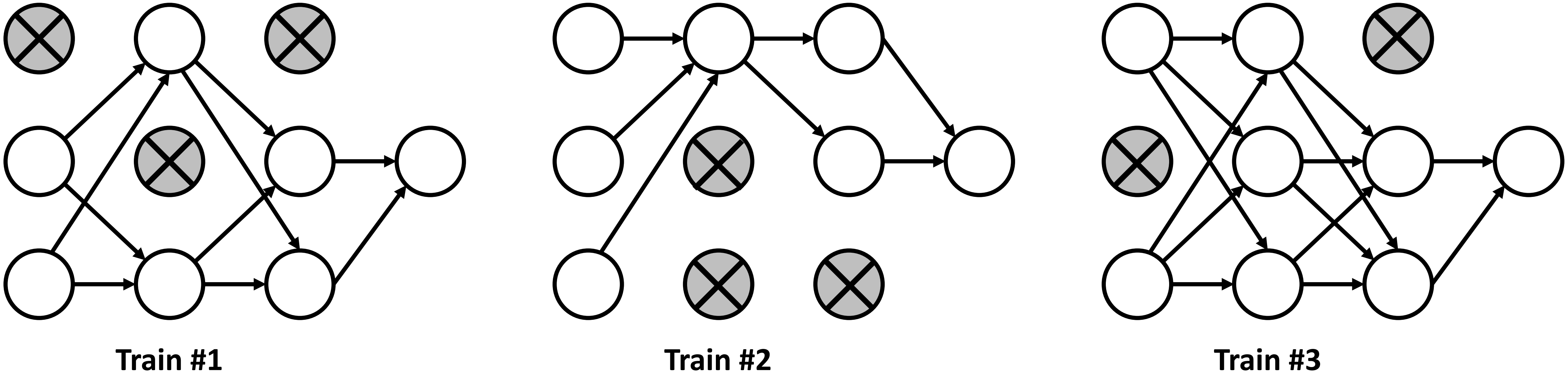
Avg. Balanced Accuracy



- Overall the average of balanced accuracy is **decreased but similar**.
- Because the balanced accuracy of Class 1 is decreased.
- The average of balanced accuracy is **80.09%**.

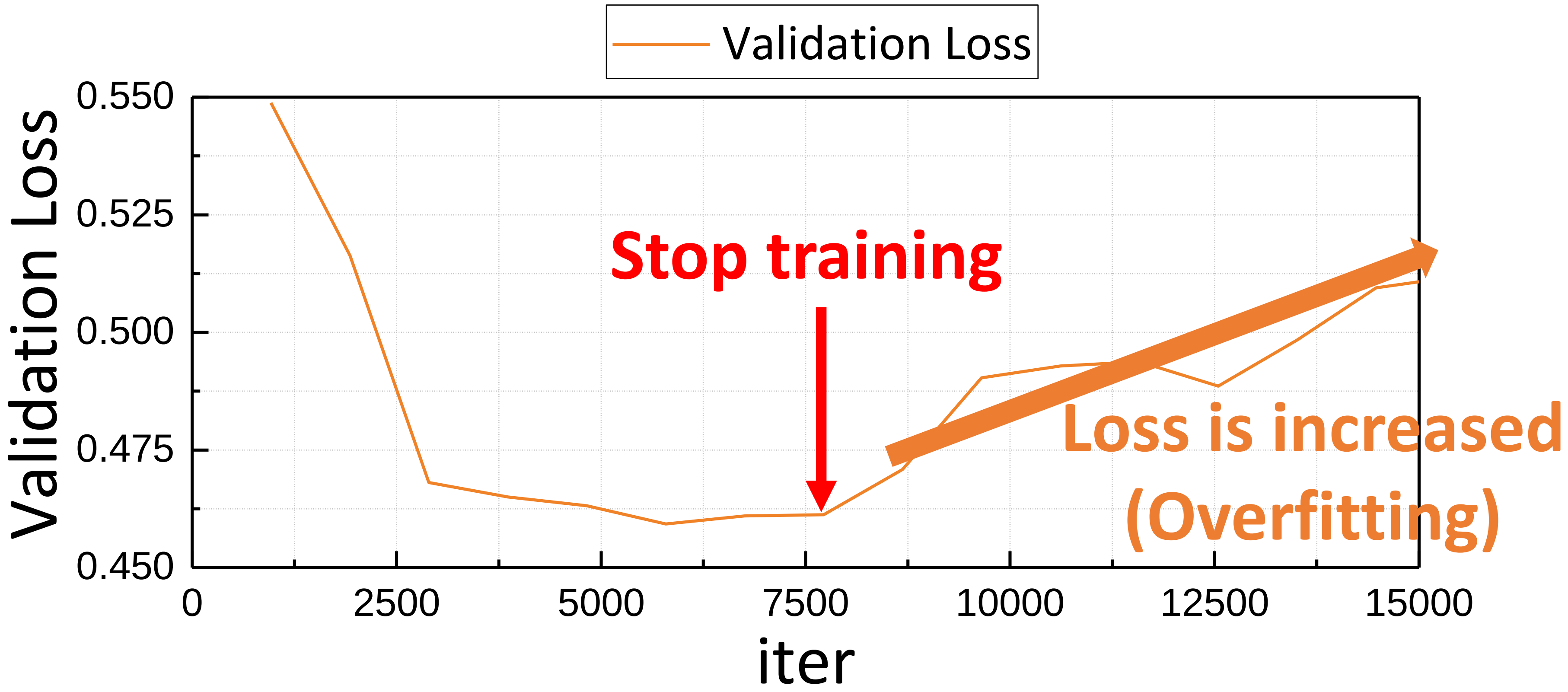
Backup

Drop-out



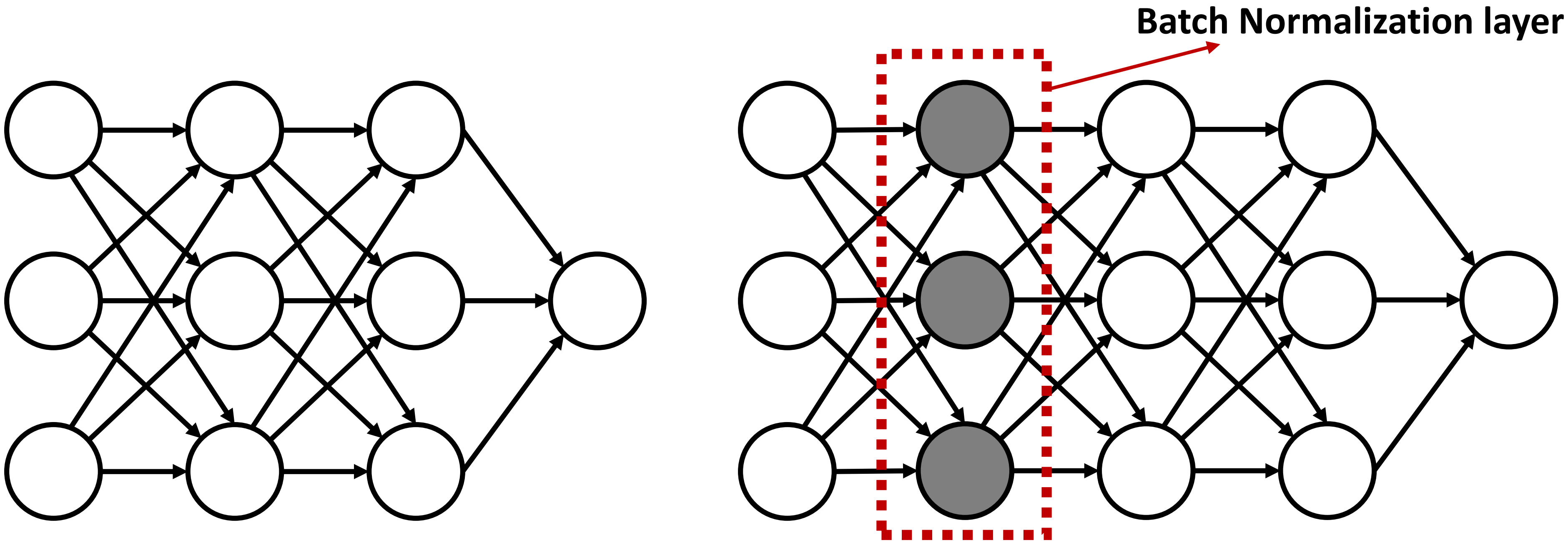
- The effect of using multiple neural network models. (performance increases)
- Prevent overfitting by reducing complexity

Early Stopping



- Stop training before overfitting occurs.

Batch Normalization (BN)

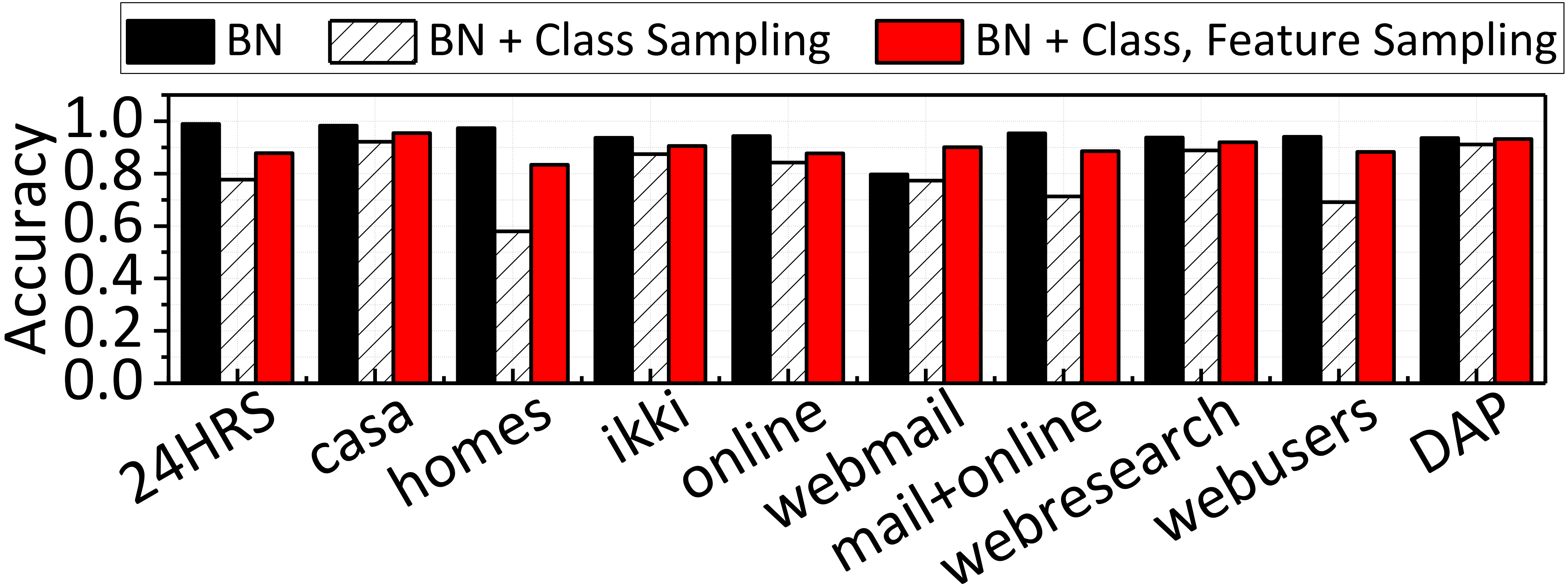


(a) Standard ANN

(b) After batch normalization

- Stable and fast training by reducing influence of parameters.
- Prevent overfitting.

Accuracy



- Applying **Feature Sampling**, the accuracy is increased.
- The average of accuracy is **89.72%**.
- The accuracy of **homes**, **mail+online** and **webusers** has improved markedly.

Recurrent Neural Network