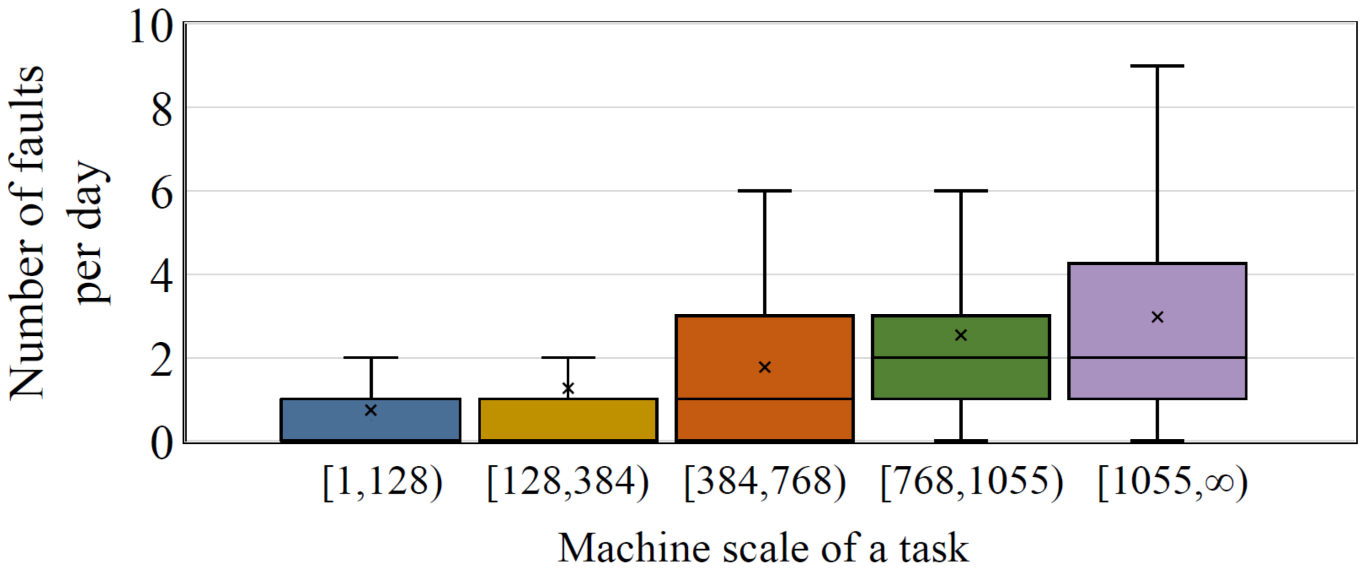
Minder: Faulty Machine Detection for Large-scale Distributed Model Training

Yangtao Deng*, Xiang Shi*, Zhuo Jiang, Xingjian Zhang, **Lei Zhang**, Zhang Zhang, Bo Li, Zuquan Song, Hang Zhu, Gaohong Liu, Fuliang Li, Shuguang Wang, Haibin Lin, Jianxi Ye, Minlan Yu

Tsinghua University ByteDance Northeastern University Harvard University

Frequent Faults in LLM Training

- **Frequent faults:** Large tasks and long durations incur more faults
- **A fault can cause a large-scale task halt:** CUDA error, NVLink error, …

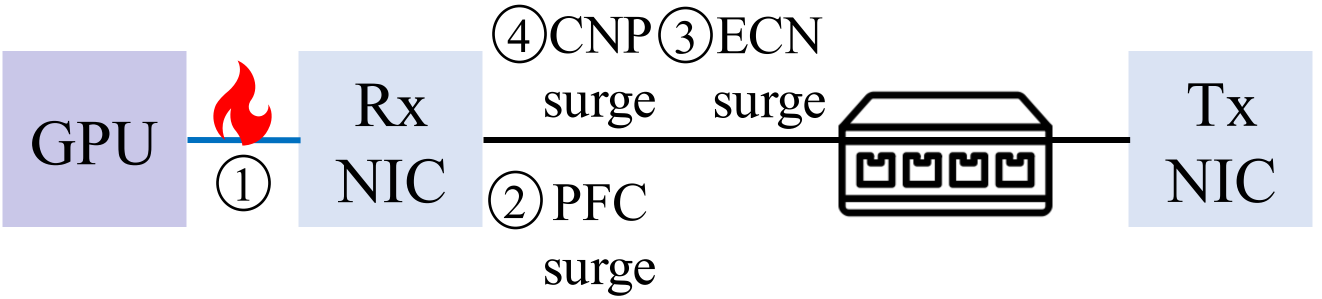


Fault frequency of tasks with different machine scale sizes

An average of two faults a day for a scale of 1000 machines in early 2024

An Example of PCIe Downgrading

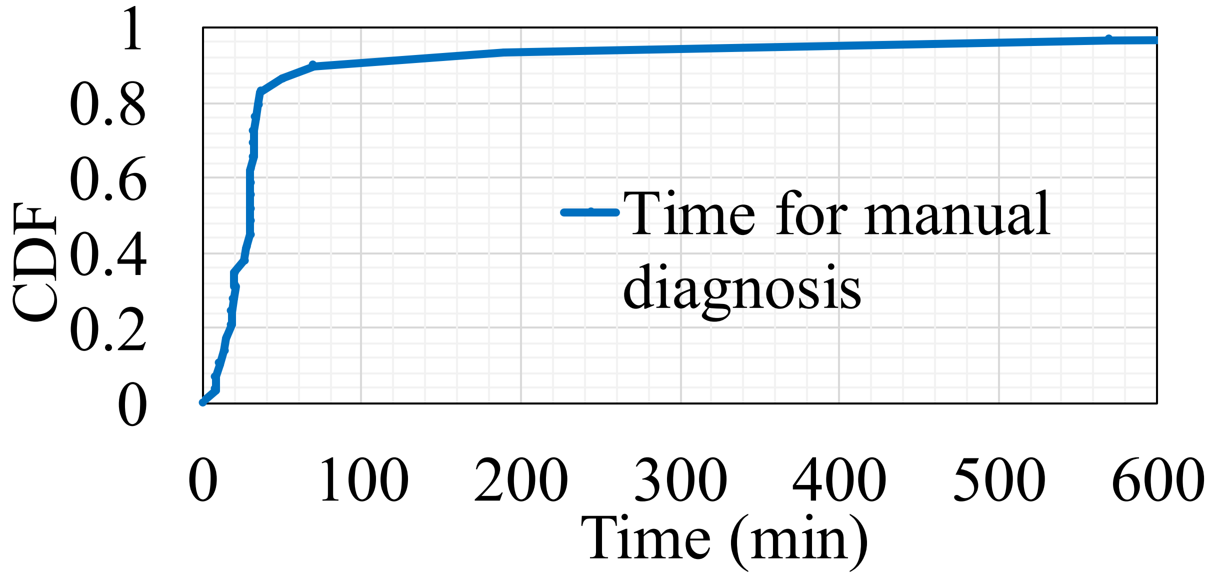
- **Multiple metrics go wrong sequentially under one failure**
  - **PCIe** in one machine downgraded from 6.4Gbps to 4Gbps ➡ slow receiving
  - **NIC buffer overflow:** NIC buffer filled up after the PCIe speed degraded
  - **PFC surge:** inter-host bottleneck caused a surge in PFC Tx to the transmitter
  - **Switch buffer overflow: ECN and CNP both increased**
  - **Throughput drop:** machine NIC throughput dropped significantly
  - **GPU underutilization:** reduced data led to declined GPU tensor core usage



- **Root cause: PCIe downgrading**
- **Multiple abnormal metrics: PFC, ECN, and CNP rates, traffic, …**

Long Time and High Costs for Fault Detection

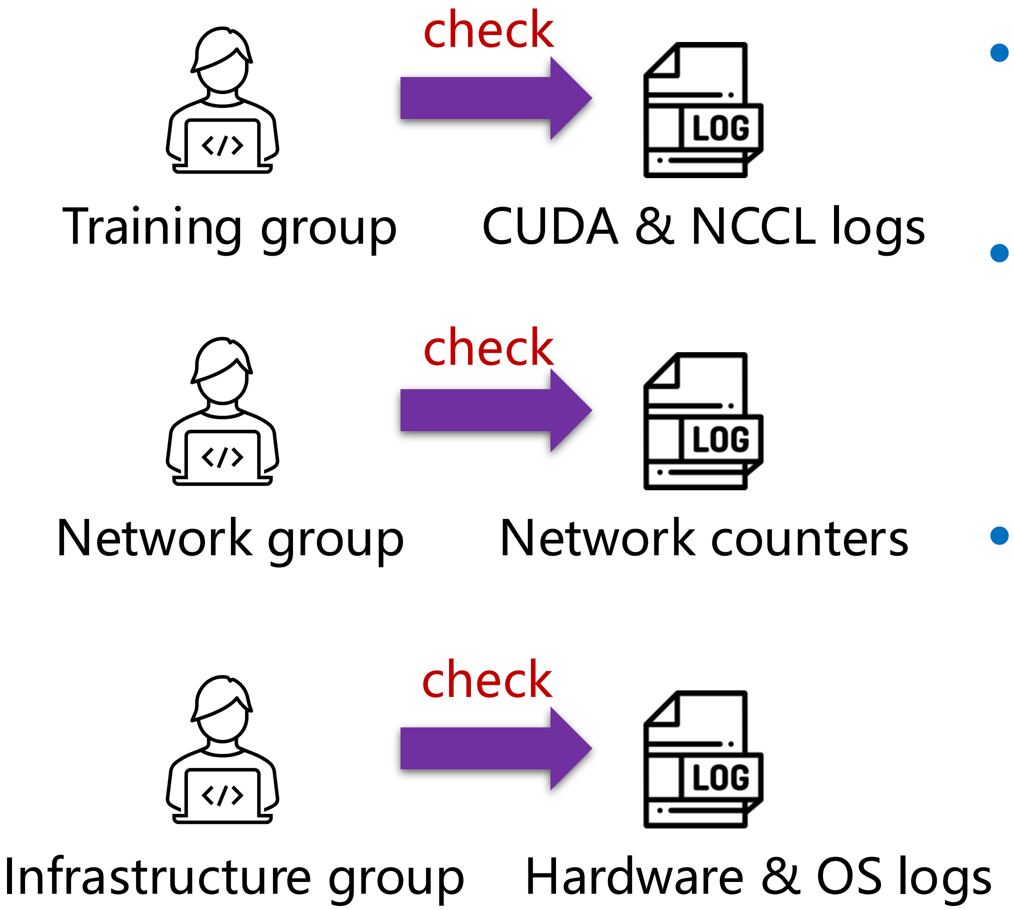
- **Multiple groups involved:** Log investigation, offline testing, …
- **The PCIe failure example:** Thousands of GPUs slow down for 40 minutes with significant cost wastes



Detection time for task diagnosis in seven months

Manual diagnosis takes more than half an hour, during which lots of GPUs are wasted

Current Approach and Limitations



Mutual log checking & Offline testing

- **Trigger of a diagnosis is not timely**
  - Only alerted once the task has stopped
- **Log content is incomplete or redundant**
  - Limited knowledge to decide which logs are useful
- **Diagnosis process is time-consuming**
  - Time to send tickets across groups
  - Each group need time to fully check logs

We need automatic and precise faulty machine detection with the logs across teams

Real-world Fault Review and Statistics

- **Hardware faults make up the majority (55.8%)**
- **Each metric indicates different types of faults with varying probabilities**

Table 1: Fault types and the proportion of instances for each fault type being indicated by a metric.

Fault type		Frequency of each fault type	Metrics					
			CPU	GPU	PFC	Throughput	Disk	Memory
Intra-host hardware faults (55.8%)	ECC error	38.9%	80.0%	65.7%	8.6%	45.7%	11.4%	57.1%
	PCIe downgrading	6.6%	0.0%	8.3%	100%	33.3%	8.3%	0.0%
	NIC dropout	5.7%	100%	100%	0.0%	100%	0.0%	100%
	GPU card drop	2.0%	75.0%	70.0%	5.0%	50.0%	20.0%	55.0%
	NVLink error	1.7%	83.3%	50.0%	16.7%	50.0%	0.0%	66.7%
	AOC error	0.9%	25.0%	25.0%	0.0%	25.0%	25.0%	25.0%
Intra-host software faults (28.0%)	CUDA execution error	14.6%	61.9%	57.1%	19.0%	33.3%	14.3%	61.9%
	GPU execution error	7.7%	50.0%	71.4%	14.3%	42.9%	21.4%	42.8%
	HDFS error	5.7%	57.1%	57.1%	0.0%	14.3%	0%	14.3%
Inter-host network faults (6.0%)	Machine unreachable	6.0%	47.4%	63.2%	0.0%	53.6%	26.3%	15.8%
Others (10.3%)	-	10.3%	-	-	-	-	-	-

Challenges 1&2: Correlating Faults&Metrics

- **Diverse faults:** Any component may fail at any time
- **No one-to-one correlation**
  - One fault may lead to many abnormal metrics
  - One abnormal metric may be caused by different faults

Table 1: Fault types and the proportion of instances for each fault type being indicated by a metric.

Fault type		Frequency of each fault type	Metrics					
			CPU	GPU	PFC	Throughput	Disk	Memory
Intra-host hardware faults (55.8%)	ECC error	38.9%	80.0%	65.7%	8.6%	45.7%	11.4%	57.1%
	PCIe downgrading	6.6%	0.0%	8.3%	100%	33.3%	8.3%	0.0%
	NIC dropout	5.7%	100%	100%	0.0%	100%	0.0%	100%
	GPU card drop	2.0%	75.0%	70.0%	5.0%	50.0%	20.0%	55.0%
	NVLink error	1.7%	83.3%	50.0%	16.7%	50.0%	0.0%	66.7%
	AOC error	0.9%	25.0%	25.0%	0.0%	25.0%	25.0%	25.0%
Intra-host software faults (28.0%)	CUDA execution error	14.6%	61.9%	57.1%	19.0%	33.3%	14.3%	61.9%
	GPU execution error	7.7%	50.0%	71.4%	14.3%	42.9%	21.4%	42.8%
	HDFS error	5.7%	57.1%	57.1%	0.0%	14.3%	0%	14.3%
Inter-host network faults (6.0%)	Machine unreachable	6.0%	47.4%	63.2%	0.0%	53.6%	26.3%	15.8%
Others (10.3%)	-	10.3%	-	-	-	-	-	-

Challenges 3&4: Hard to Define Anomaly

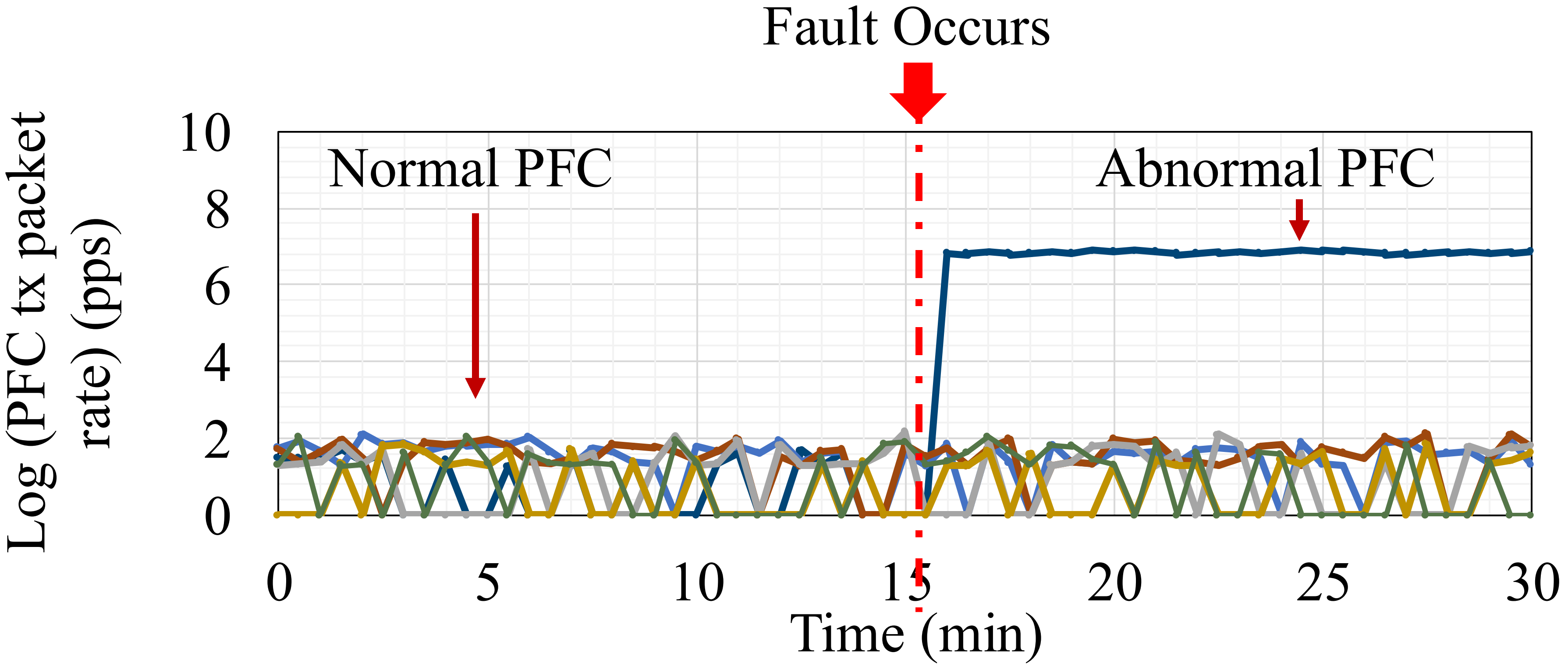
- **Task-dependent anomaly**
  - E.g., GPU temperature of 70 Celsius is **abnormal** for 1350MHz GPU Clock
  - But **normal** for 1800MHz GPU Clock
  - Different PFC thresholds for different scale of tasks

- **Noises in time-series metrics**
  - Jitters, inaccurate sensors, faulty data collection, and network interruptions
  - Short-term noises may be misleading in fault detection

Insight 1: Machine-level Similarity

– addressing challenges 1 & 2

- **Machine learning is highly parallelized across machines**
- **Faulty machine exhibits dissimilar patterns in monitoring metrics during parallel training**

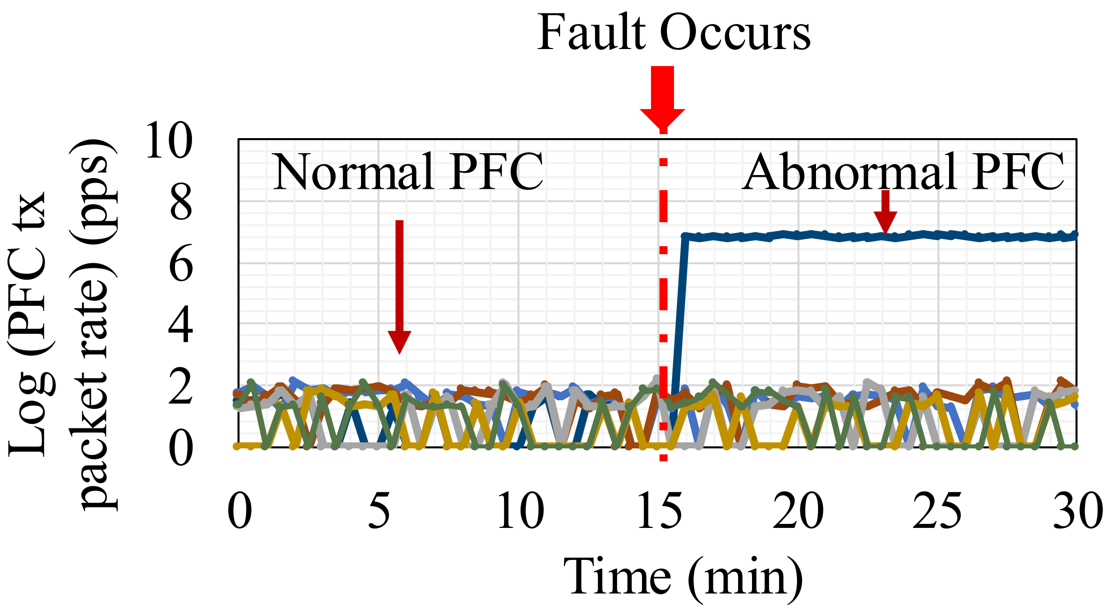


PFC tx packet rate before and after a fault occurs

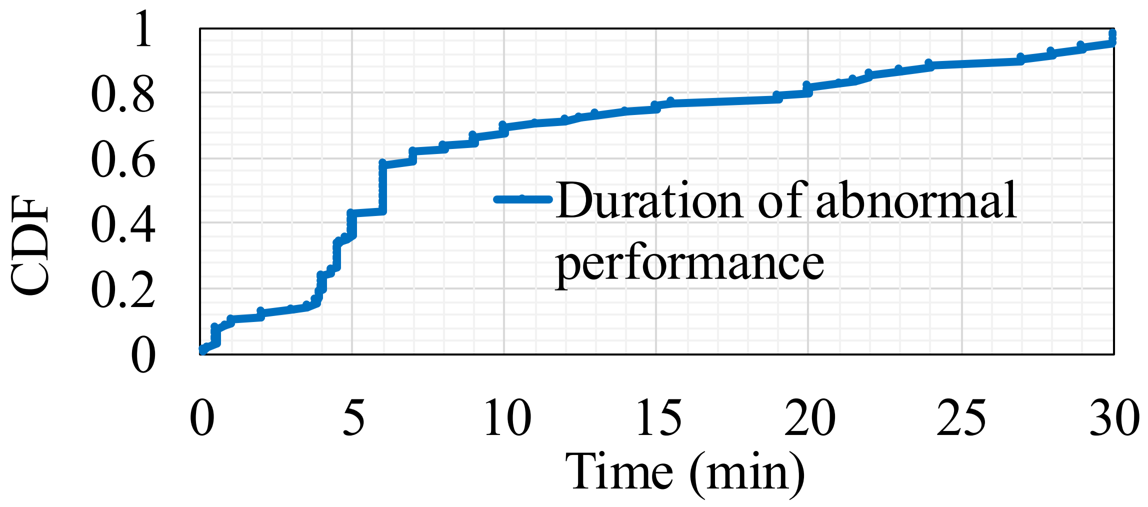
Insight 2: Machine-level Continuity

– addressing challenge 4

- **Machine learning is repetitive across iterations**
- **Abnormal metrics typically persist for some time**



PFC tx packet rate before and after a fault occurs

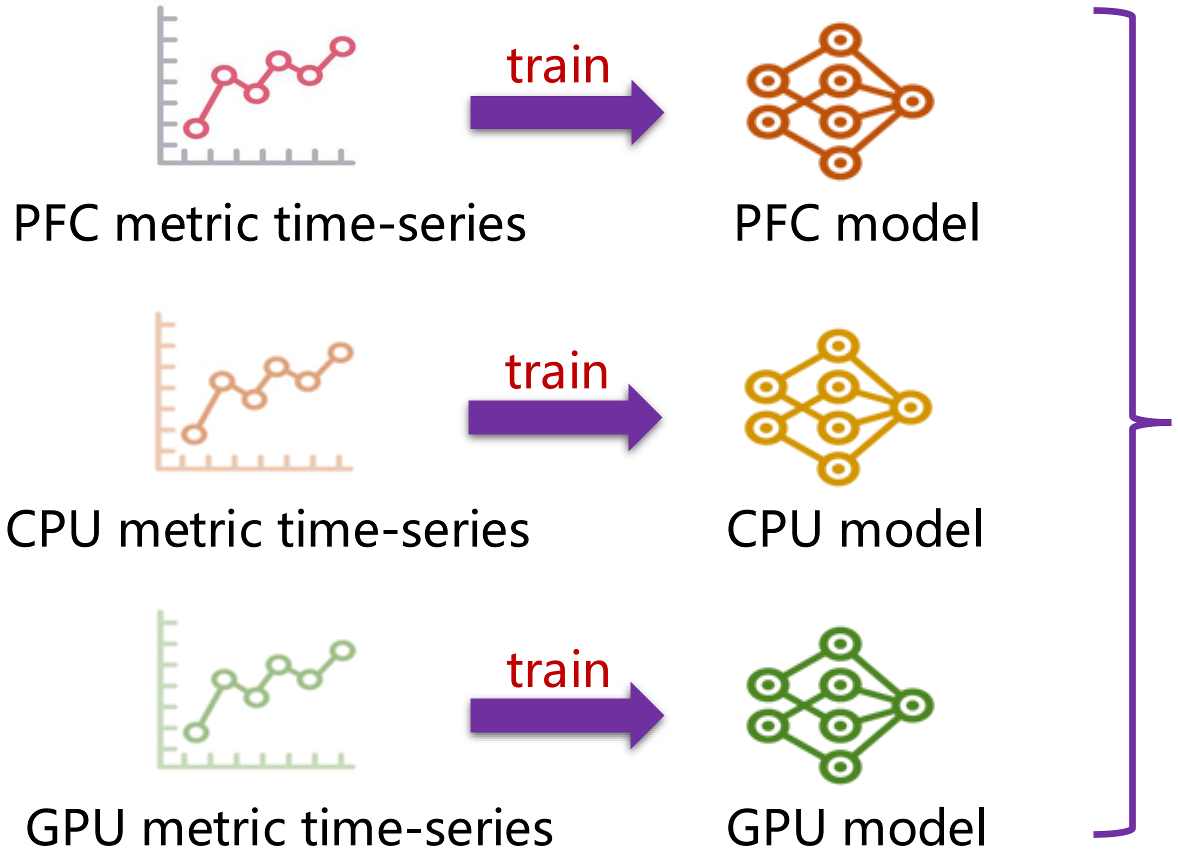


Duration of abnormal performance following a fault

Insight 3: Separated Models

– addressing challenge 2 & 3

- **Separated models for each metric to differentiate (ab)normal behaviors**

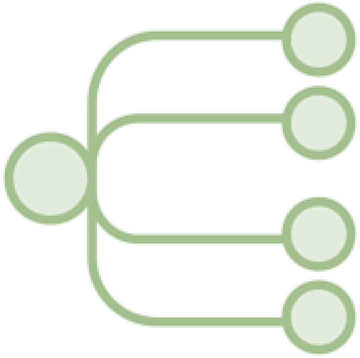


Monitoring metrics exhibit an "or" correlation

Overview

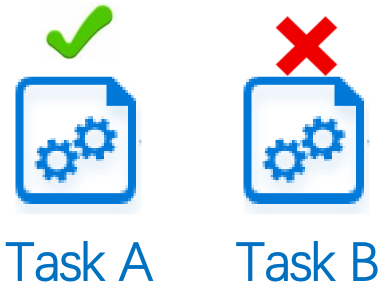
Diverse fault types

Diverse faults occur at any component in a machine



One-to-many correlation

Monitoring metrics exhibit an "or" correlation



Task-dependent anomaly

The abnormality of monitoring metrics is task-dependent

Noises

Noises exist in time-series monitoring data

Comprehensive monitoring

Individual models for each monitoring metric

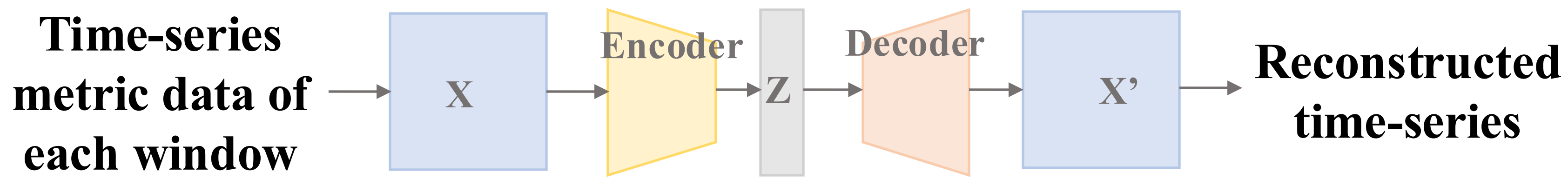
Denoising for accurate detection

Minder

- **Per-machine time series of metrics as input, deep learning and comparison to identify faulty machines**
- **Two key steps:**
  - VAE-based per-metric models → denoising and compression
  - Similarity & continuity-based check → automatic and precise detection

VAE based Per-metric Models

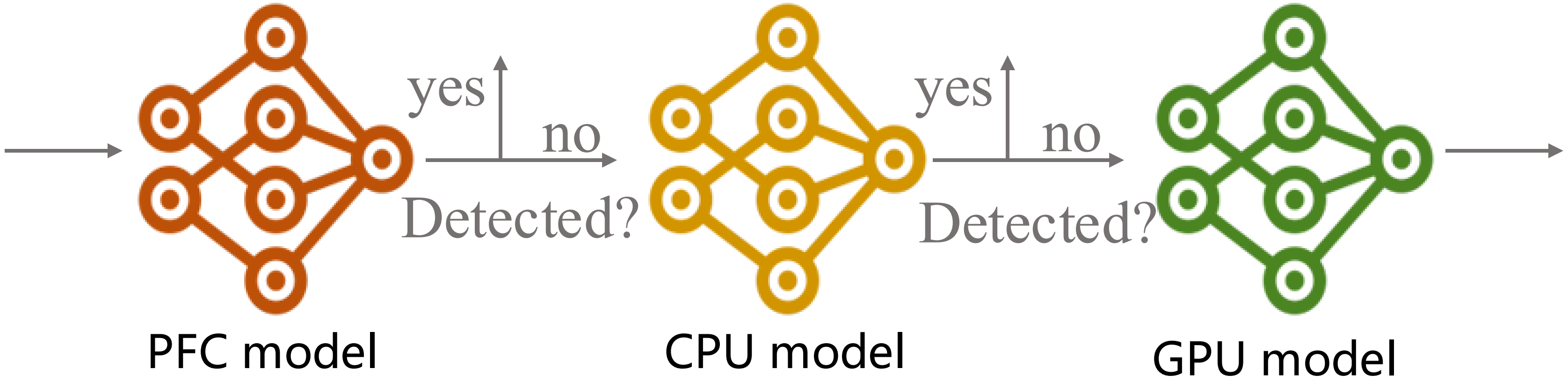
- **Unsupervised LSTM-based Variational Autoencoder (VAE):**
  - **Unsupervised:** hard to label task-dependent faults
  - **VAE:** enhance the accuracy and robustness of anomaly detection w/o labels
    - **Learn vector distribution**
    - **Remove noises** by reconstructing into new dimensions
    - **Compress** a high-dimensional features into a smaller dimension space
  - **LSTM** as encoder and decoder for time series data



VAE structure

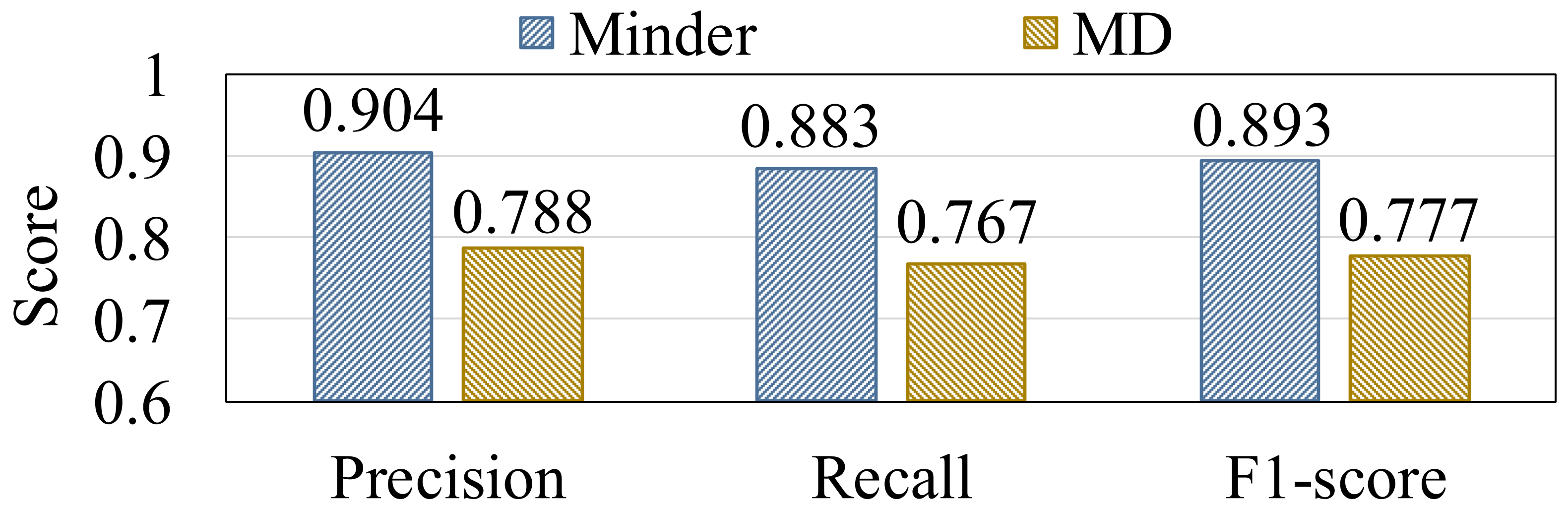
Similarity & Continuity-based Check

- **Similarity check**
  - Z-score for each metric j: Computes the machine i differing from the overall behavior across machines with a threshold
- **Continuity check**
  - Detected in consecutive time windows
- **Decision tree to order metrics based on their Z-scores**
  - Repeat the detection with each model until a faulty machine is detected



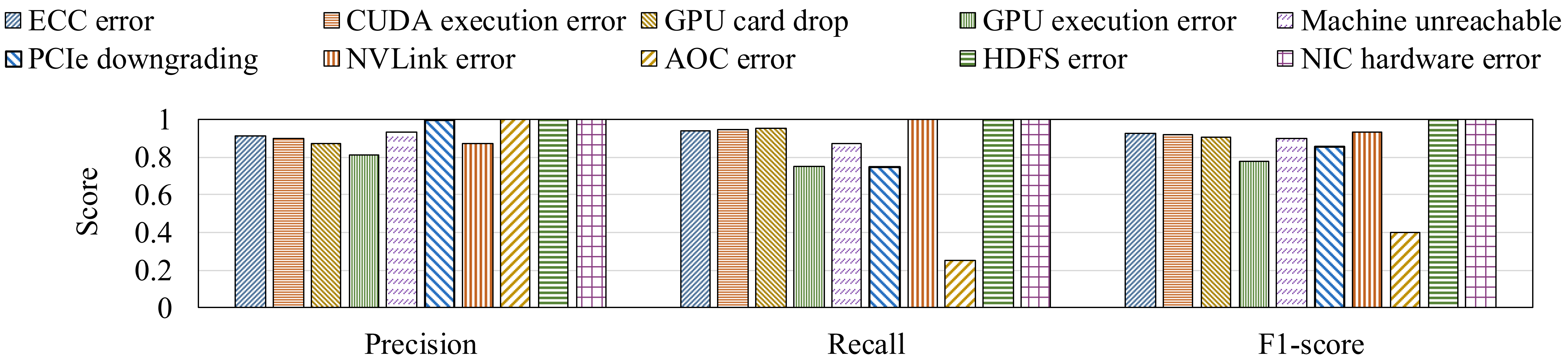
Deployment and Evaluation

- **Implemented as an always-on backend service in ByteDance**
  - Running tasks with a cluster of **1000+** machines
  - Reducing the detection time by **99%** in our dataset
- **Precise comparison with a baseline**
  - Mahalanobis Distance (MD): variable correlations, feature PCA, …
  - Minder outperforms MD by using VAE for denoising and extracting data patterns for a better distance calculation.



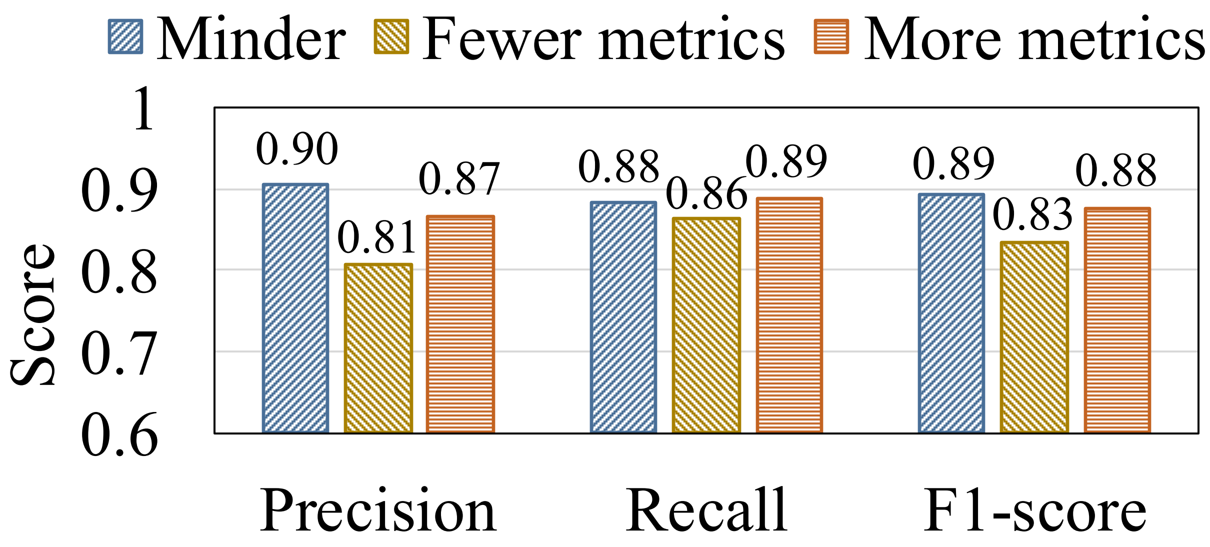
Deployment and Evaluation

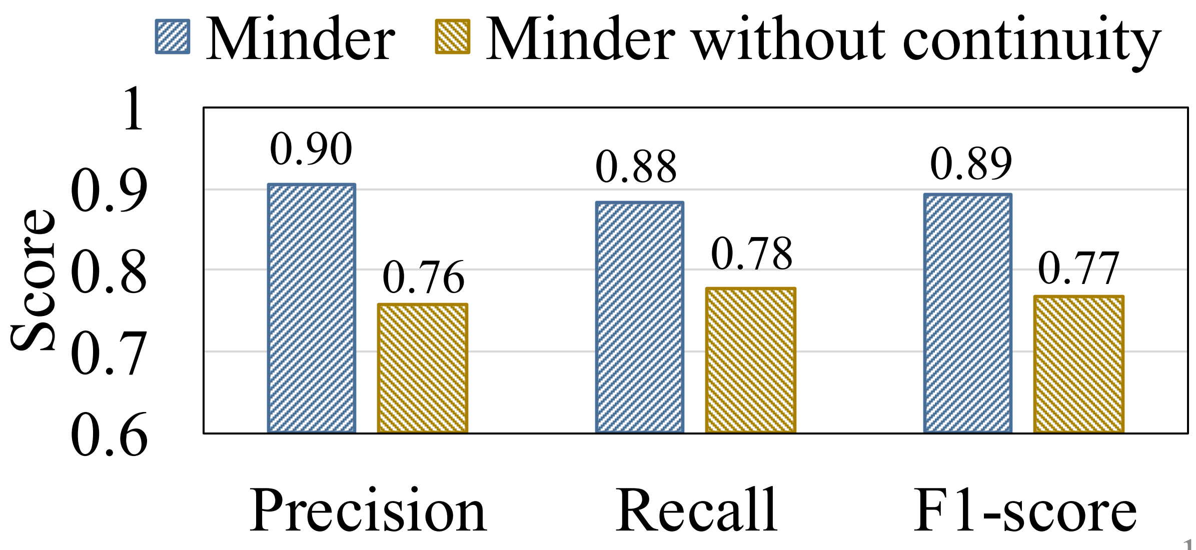
- **Accuracy for various fault types**
  - **> 90% precision** for many failure types
  - CPU and GPU related errors are easy to detect
  - PFC, CPU, and GPU models are enough for most faults



Deployment and Evaluation

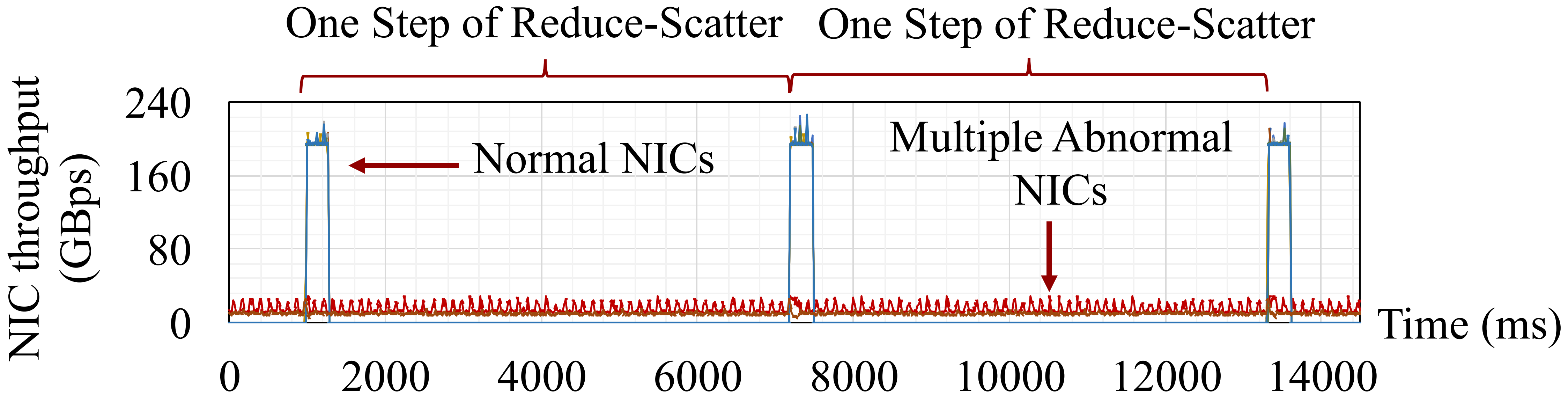
- **Comparison with different metric selections**
  - More metrics introduce mutual interference
  - Fewer metrics undermine outlier detection capacity
- **Accuracy with/without continuity**
  - More false alarms without continuity





Multiple Concurrent Faulty Machines

- **Two key factors**
  - **Faulty machine scale ratio:** more faulty machines impact more groups; faster propagation across machines
  - **Granularity of monitoring data:** the dissimilar pattern being overlooked due to coarse-grained monitoring



Millisecond-level NIC throughput PCIe downgrading injection on two NICs

Experience

- **Integration with other monitoring tools**
  - Other monitoring tools used along: DCGM, EUD, RDMA traffic alerts, switch monitoring, R-Pingmesh[SIGCOMM'24], ...
- **Minder's generality**
  - Flexible in **data granularity**: second-level, millisecond-level, ...
  - Flexible in the **metric spectrum**: out-of-band hardware counters, AOC counters, …
- **Minder's robustness of other faults**
  - As long as the monitoring data presents discernible **dissimilarities**
- **Not all failed tasks have the right label**
  - Temporary performance fluctuations and jitters

Conclusion

- **Frequent failures in large-scale distributed training**
  - Faulty machine detection is critical for saving labor and resource costs
- **An automatic system to tackle faulty machine detection**
  - Machine-level similarity and continuity
  - Unsupervised per-metric models
- **Fast and accurate detection in production environments**
  - Minder reduces the detection time by 99% with a precision of 0.904 and F1-score of 0.893 on average

Thank you for listening!

- *Check our paper:* Minder: Faulty Machine Detection for Large-scale Distributed Model Training

清華大學 Tsinghua University

ByteDance

HARVARD UNIVERSITY