Reliability and Performance Evaluation of Two-input Machine Learning Systems

Kazuya Wakigami, Fumio Machida, Tuan Phung-Duc
University of Tsukuba

28th IEEE Pacific Rim International Symposium on Dependable Computing

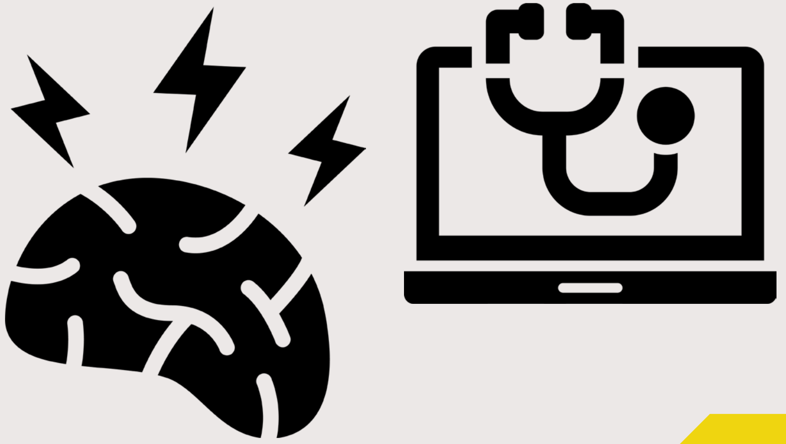
Outline

1. Introduction
2. Related Work
3. Two-input Machine Learning Systems
4. Experiment Procedure
5. Empirical Results
6. Comparison with Simulation Results
7. Conclusion

Introduction

Introduction(1)

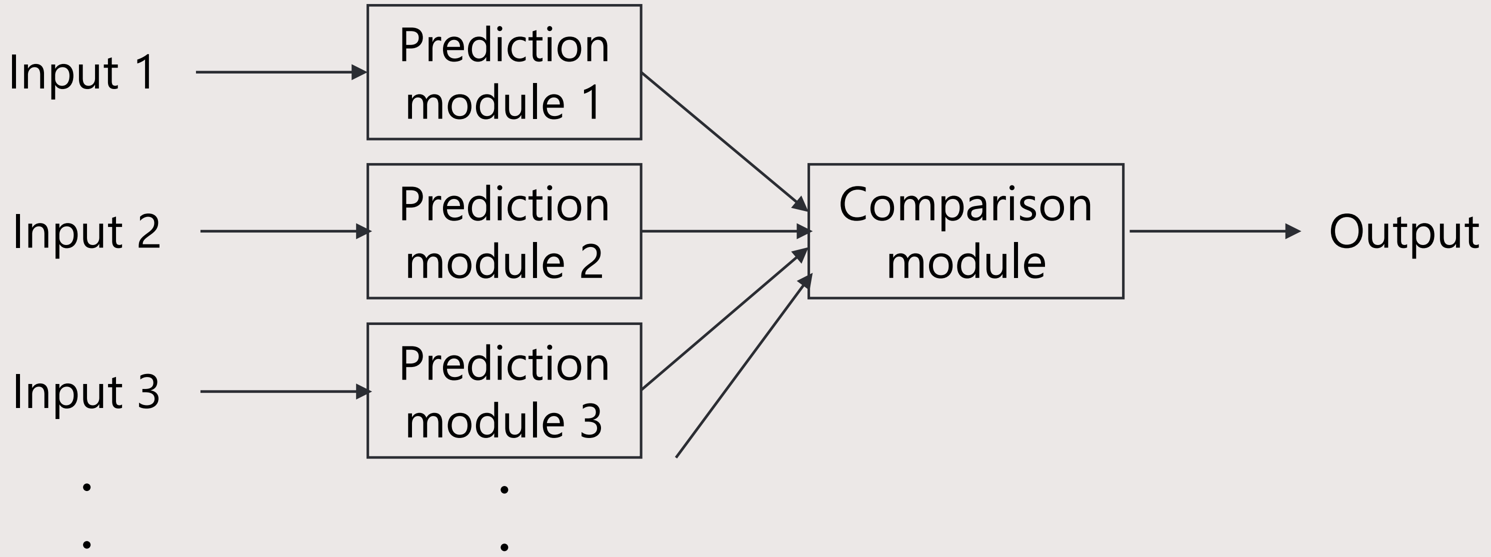
- ML (Machine Learning) models have been widely used.
- Applications of MLs are expanding in the fields requiring safety and high reliability, such as medical image diagnosis and autonomous vehicles.



- Prediction errors may cause serious problems.

Introduction(2)

- N-version MLS (Machine Learning System) [1]
  - A redundancy architecture
  - Use more input and/or ML modules.
  - Decrease throughput performance.



[1] F. Machida, "N-version machine learning models for safety critical systems," Proceedings of DSN Workshop on Dependable and Secure Machine Learning, pp. 48-51, 2019.

Introduction(3)

- Two-input MLS
  - One architecture of the N-version MLSs.
  - System output are determined by two prediction results for two input.

Image 1

Image 2

Prediction result 1 for Image1
Prediction result 2 for Image2

One system output

Related Work

Related Work(1)

- Throughput performance of two-input MLSs is evaluated in [2].
  - If the arrival rate cannot be changed and the processing speed is sufficiently large, the parallel type has higher throughput than the shared type.

[2] Y. Makino, T. Phung-Duc, and F. Machida, "A queueing analysis of multi-model multi-input machine learning systems," Proceedings of The 4th DSN Workshop on Dependable and Secure Machine Learning, 2021.

- Response time and power consumption of two-input MLSs is evaluated in [3].
  - Shared type architecture has lower response time and energy consumption than parallel type architecture.

[3] S. Nishio, Y. Makino, T. Phung-Duc, and F. Machida, "Performance Analysis of Energy-Efficient Reliable Machine Learning System Architectures," http://dx.doi.org/10.2139/ssrn.4431918, 2023.

Related Work(2)

- The latency and energy consumption of the object detection model are evaluated in [4].

  [4] J. Lee, P. Wang, R, Xu, V. Dasari, N. Weston, Y. Li, S. Bagchi, and S. Chaterji, Virtuoso: Video-based Intelligence for real-time tuning on SOCs, ACM Transactions on Design Automation of Electronic Systems, Association for Computing Machinery New York, NY, United States, 2022.

- Accuracy, inference time, and energy consumption of the image classification tasks are analyzed in [5].

  [5] A. Canziani, A. Paszke, E. Culurciello, "An analysis of deep neural network models for practical applications," arXiv preprint arXiv: 1605.07678, 2016.

Difference with Related Work

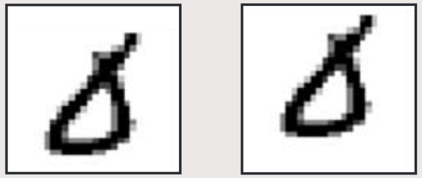
- The performance of two-input MLSs has been theoretically investigated in the previous study using queueing analysis.
- However, the existing studies have not verified the performance characteristics of two-input MLSs with real MLSs.

- In our study, we implement two-input MLSs and empirically investigate the performance characteristics of real MLSs.

Two-input Machine Learning Systems

Two-input MLSs(1)

- In the previous work [2] and [3], the parallel type and the shared type architectures are theoretically evaluated.
- If the inference results are not matched, the system can find that at least one of the results is wrong, and hence, an incorrect system output can be suppressed.
- In the case of image classification task using number images:



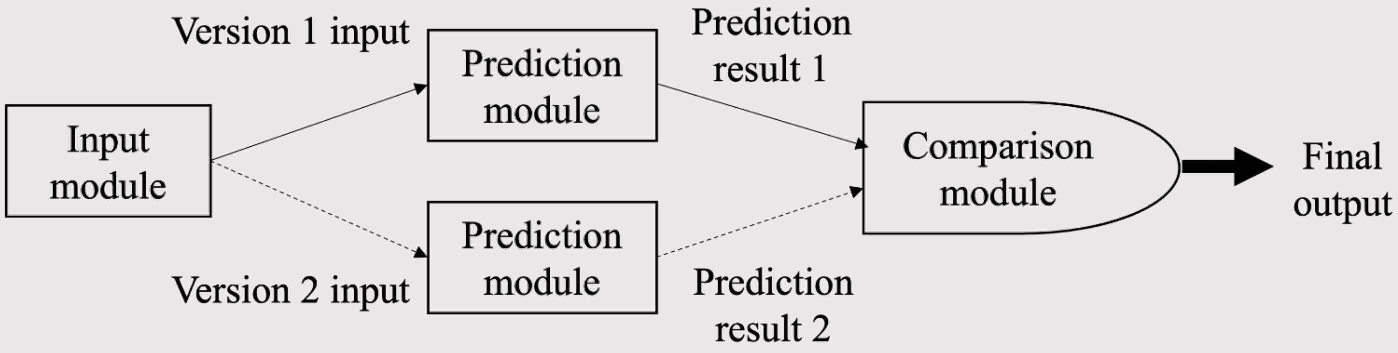
Input 1 (Original) Input 2 (Shifted)

→ 0 6 → At least one result is wrong.

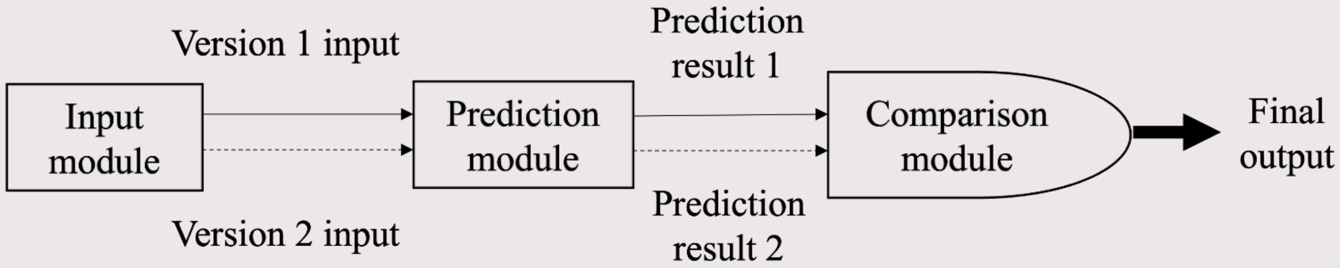
Result 1 Result 2

Two-input MLSs(2)

- In our work, we focus on two architectures of two-input MLS

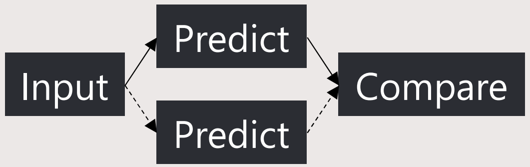


Parallel type



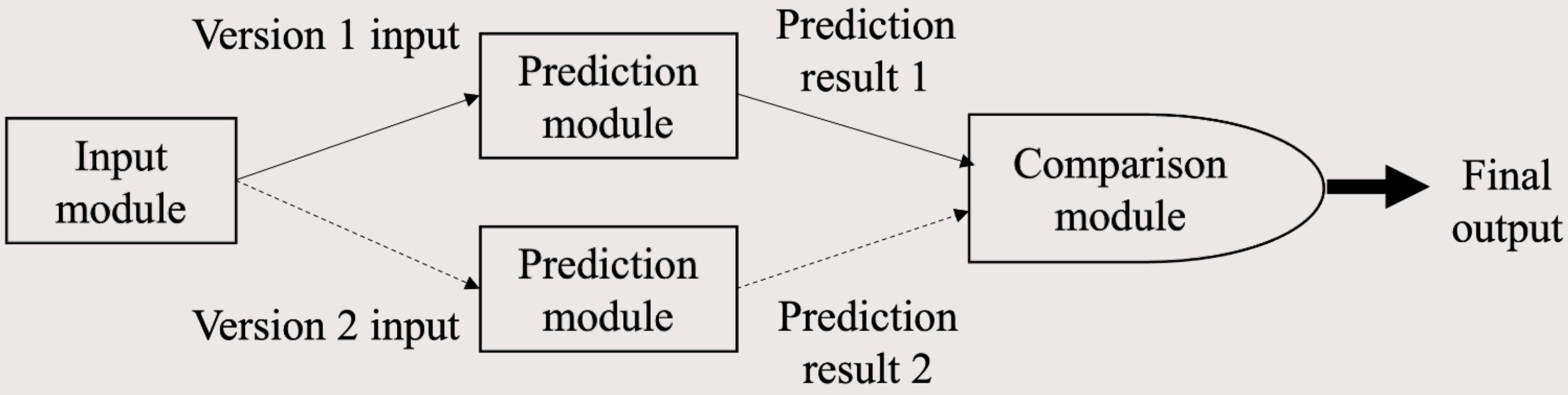
Shared type

Parallel type architecture

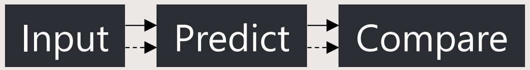


Parallel type

- Version 1 and Version 2 input are sent to different Prediction modules.
- All the inference results are sent to the Comparison module that decides the final output of the system.

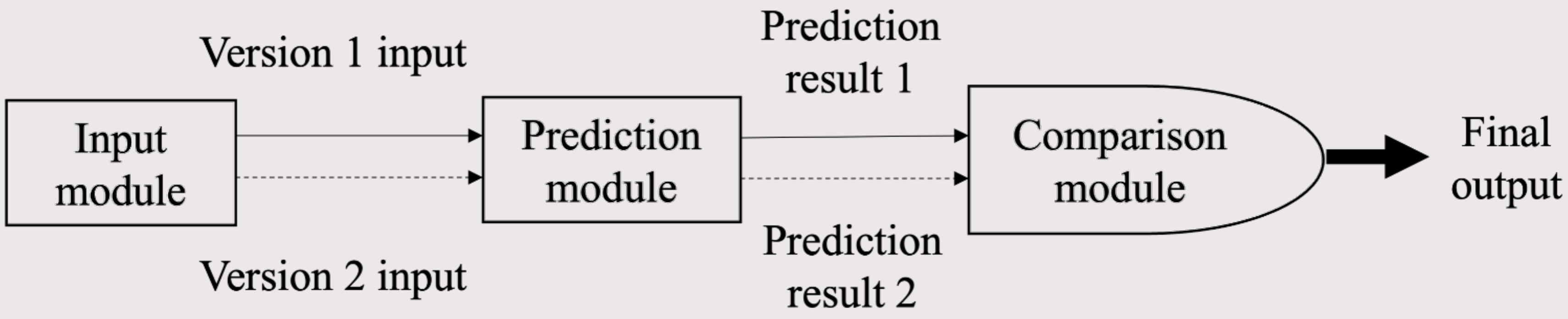


Shared type architecture



Shared type

- Version 1 and Version 2 input are sent to the same Prediction module.
- All the inference results are sent to the Comparison module that decides the final output of the system.

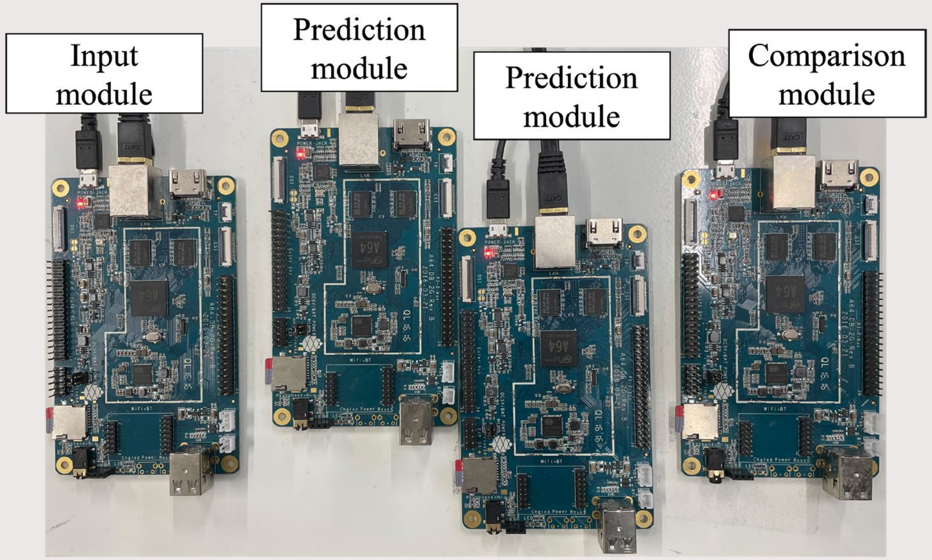


Experiment Procedure

PINEs

- We Implement the experiment system composed of four PINE A64s.

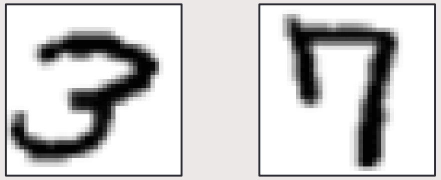
- Specifications
  - CPU: Quad-core ARM Cortex-A53 Processor@1152Mhz
  - RAM Memory: 2GB
  - OS: Armbian 22.05.3 Focal



ML model

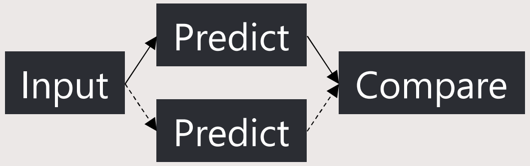
- We consider an image classification task for MNIST dataset as an ML model.

- Google colaboratory, PyTorch as an ML framework
- CNN (Convolutional neural network) trained with 60,000 MNIST training data.
- ReLU (Rectified Linear Unit) as the activation function
- Cross-entropy Loss as the loss function
- Adam as the optimization function.

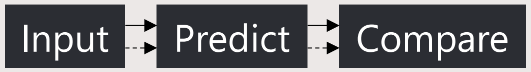


Example of the MNIST dataset

Performance measurements



Parallel type



Shared type

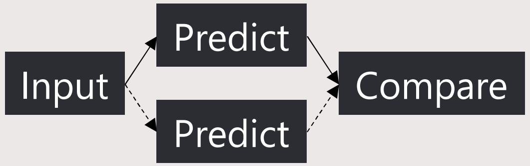
- We built two experiment systems,
  parallel and shared type architecture, using PINEs and ML model.
- The input data are sent in two interval patterns.
  - Following the Poisson distribution (arrival rate $\lambda 1 = \lambda 2 = 10$).
  - Constant time intervals (0.1 seconds).
- The maximum buffer size of the Prediction module is set to $K = 80$.

Service time measurements

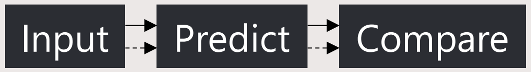
- We also measure the service time of the Prediction module in the MLSs.
- Service time indicates the time required for module processing (i.e., ML model inference).
- Send input data 10,000 times for each architecture.

Empirical Results

Correct output ratio - Empirical



Parallel type



Shared type

- The correct output ratio is computed by dividing the number of correct output by the number of output.
- Two-input MLS can improve the correct output ratio by exploiting data diversity, as expected from the theoretical results [3].

Two-input MLS

Table 1. Mean correct output ratio

	Parallel type	**Shared type**	**1-ver.**
Poisson distribution	0.9958	0.9988	0.9678
Constant interval	0.9973	0.9953	0.9676

Comparison ratio - Empirical

- Comparison ratio:

$$\frac{\text{the ratio of the number of comparison processes}}{\text{the total number of data pairs sent from the Input module}}.$$

- In the case of the data input interval following the Poisson distribution, the comparison ratio of the shared type architecture is 66.39 % ($\fallingdotseq \frac{2}{3}$).

Table 2. Mean comparison ratio

	Parallel type	**Shared type**	**1-ver.**
Poisson distribution	0.9943	0.6639	0.9997
Constant interval	0.9982	0.9996	0.9993

Response time(1)

- Two-input MLSs have longer response time.
- Response time (parallel, Poisson) is significantly affected by the waiting time in the buffer due to the randomness of the data arrival.
- Response times (constant interval) are shorter than the response times in the Poisson distribution case.

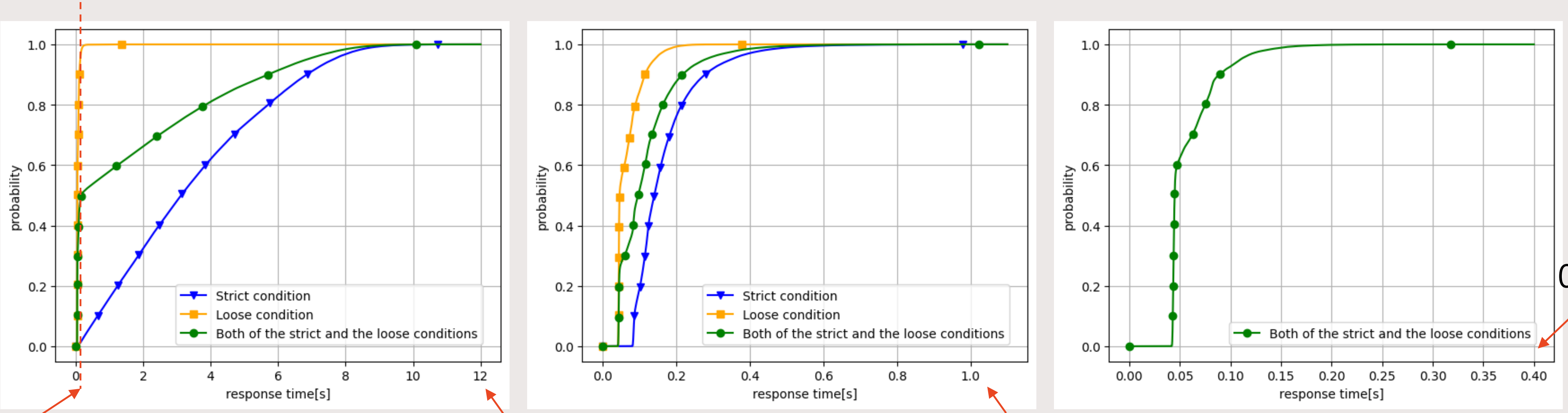
Two-input MLS

Table 3. Mean response time (Both conditions)

	Parallel type	**Shared type**	**1-ver.**
Poisson distribution	1.7722	0.1176	0.0585
Constant interval	0.3101	0.0925	0.0437

Response time(2)

- Parallel type has large range of values about 0s to 10s, and about 50% is shorter than 0.04s.
- Shared type also has larger range 0s to 0.5s than single version.



0.04s (a) Parallel type 12s (b) Shared type 1s (c) Single version

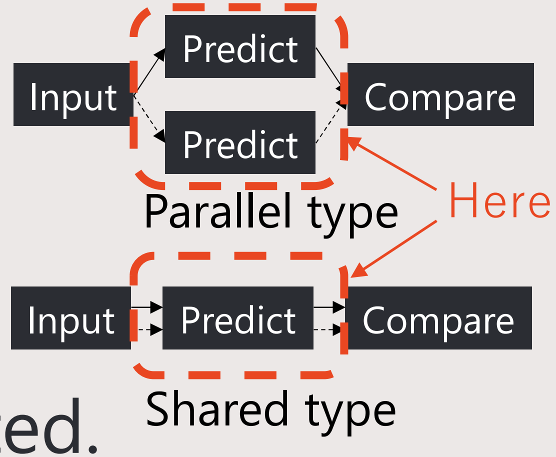
Energy consumption

- Energy consumption is measured in every second.
- The mean energy consumption of the shared type architecture is 25.52 % smaller.
- This is due to the difference in the number of machines used for each architecture.

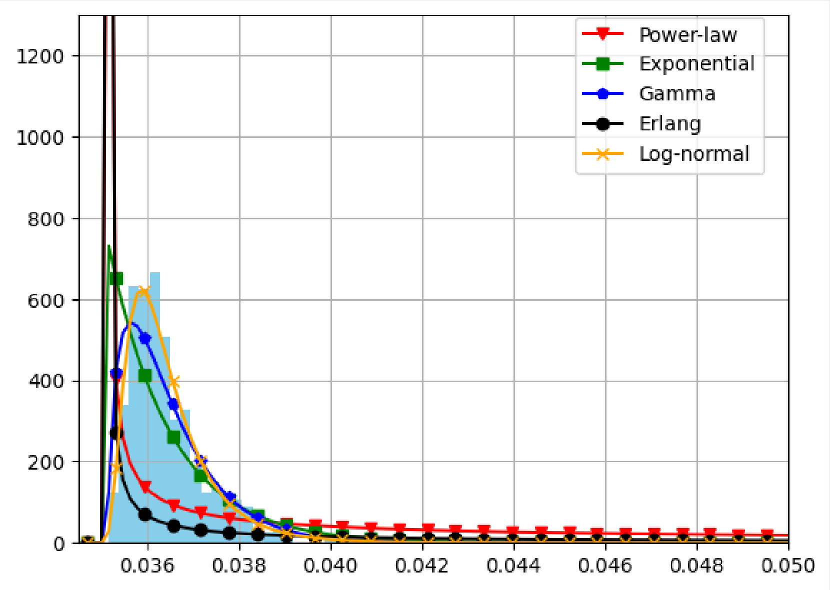
Table 4. Energy consumption

	Energy consumption [W]
Parallel type	12.03
Shared type	8.96

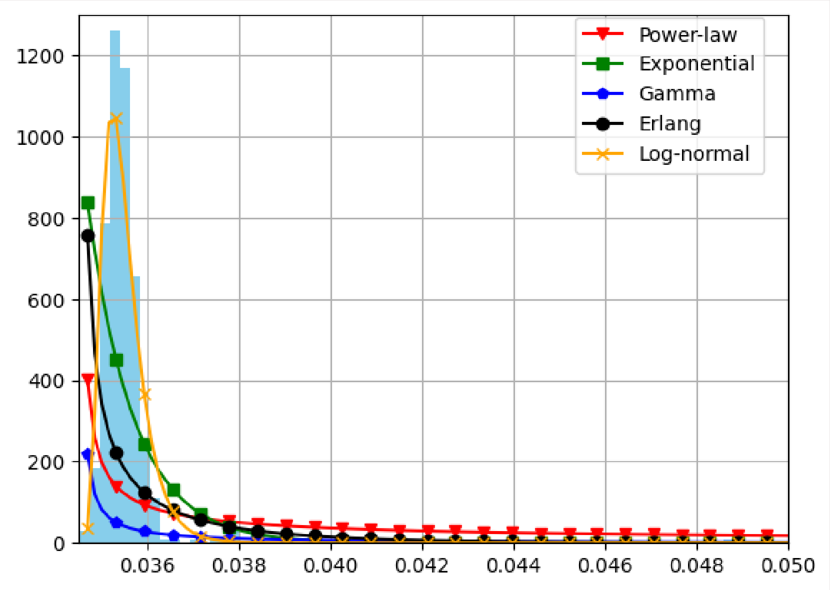
Inference time distribution



- We measure the inference time and use fitter library.
- The log-normal distribution, yellow line (×) is well fitted.



(a) Poisson distribution



(b) Constant distribution

Comparison with Simulation Results

Comparison with Simulation Results(1)

- We compare the response time measured in the real MLSs and the response time obtained by a simulation program.
- The simulation program is developed using the queueing model [3].
- Configuration of the simulation program
  - Parameters are set to be consistent with the empirical system.
  - Inference time distribution is different.
    (simulation: exponential, empirical: log-normal).

Comparison with Simulation Results(2)

- The mean response time of empirical results is shorter in the parallel type architecture.
- In the shared type architecture, the result become the opposite.
- The empirical minimum response times are longer for both architectures.

Table 5. Comparison of the response time

(a) Parallel type architecture [s]

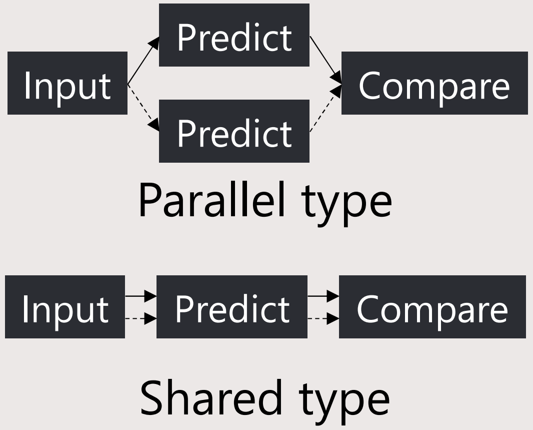
	Simulation	Empirical
Mean	2.061	1.772
Standard deviation	6.748	5.537
Minimum	0.0000468	0.0422
Maximum	10.500	11.160

(B) Shared type architecture [s]

	Simulation	Empirical
Mean	0.111	0.118
Standard deviation	0.0099	0.0072
Minimum	0.0000257	0.0411
Maximum	0.857	1.040

Conclusion

Conclusion(1)



- We conducted experiments to evaluate the reliability, performance, and energy consumption of the MLS in the parallel type and the shared type architectures.
- The shared type architecture MLS has a lower energy consumption and a shorter response time.
- The parallel type architecture is preferable in terms of reliability since the shared type architecture reduces the throughput.

Conclusion(2)

- We compared our empirical results and the results of the simulation program [3].
- The response time of the empirical result is shorter. This is due to the difference in the distribution of the service time of the Prediction module.
- The service time distribution of the ML module fits better with the log-normal distribution