High-Availability Machine Learning Systems: N-version Architecture and Rejuvenation

Fumio Machida

University of Tsukuba

International Workshop on Advanced Intelligent Software Applications: AISQ2025

Machine Learning Systems

- Many systems involve ML and AI components

Autonomous vehicle

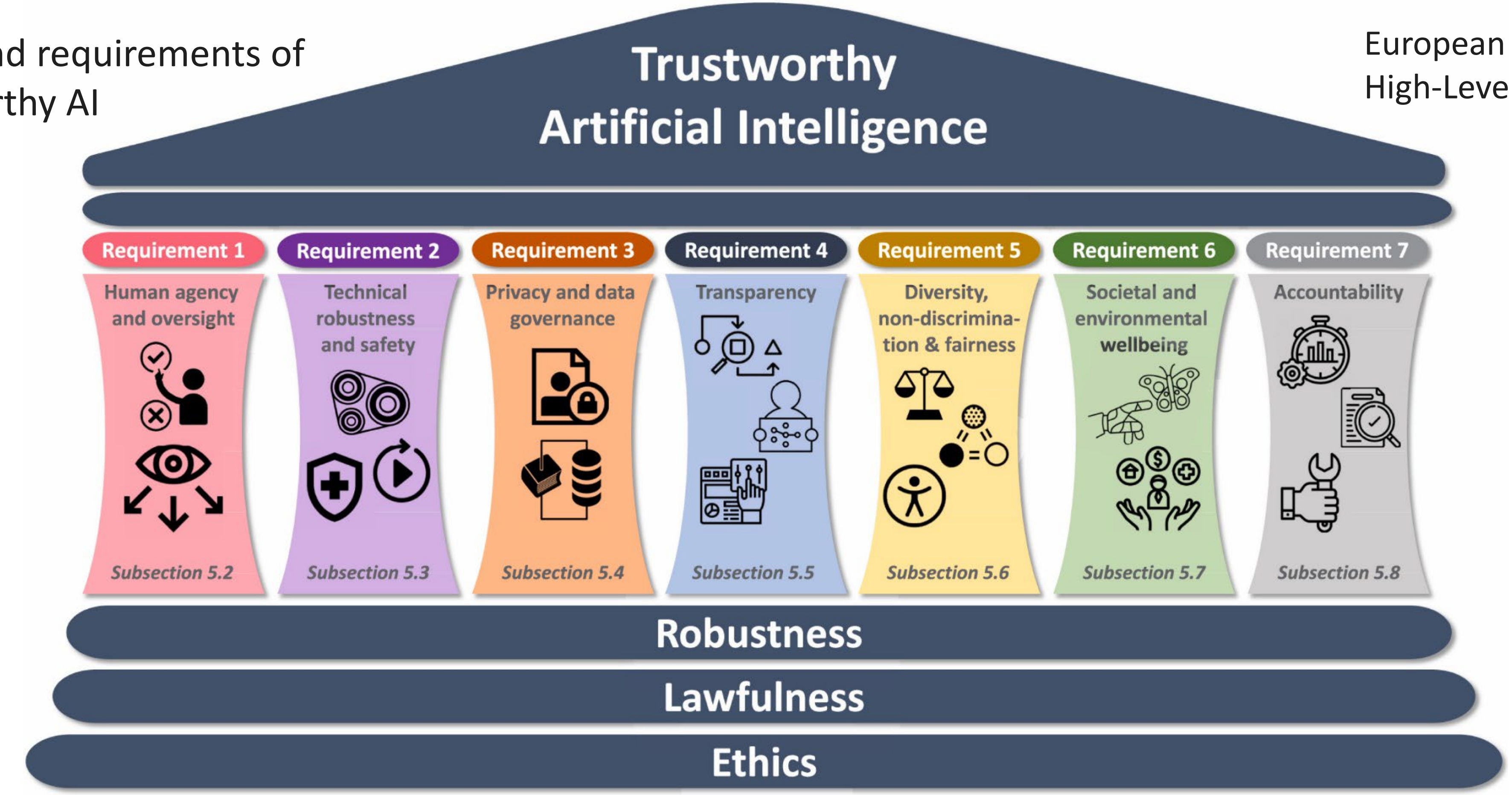
Voice assistant device

Factory automation robot

Toward Trustworthy AI Systems

Pillars and requirements of Trustworthy AI

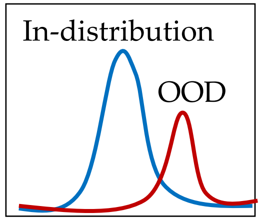
European Commission
High-Level Expert Group on AI C.

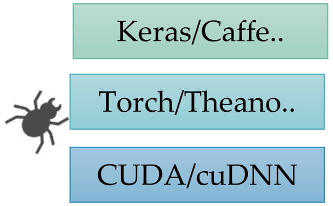


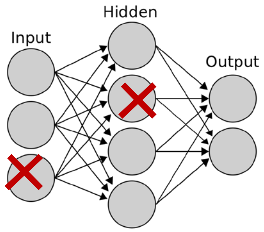
Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation

ML system reliability

- Various threats to ML system reliability
- ML model mispredictions
  - Out-Of-Distribution
  - Adversarial Example
- Software and hardware faults
  - Software bugs
  - Transient memory errors（Soft Error)







Undesirable consequences

- Failures of ML components adversely impact society

Tesla in self-driving mode causes 8 vehicle crashes

https://bit.ly/3m9kJ8b

Facial recognition technology jailed a man for days

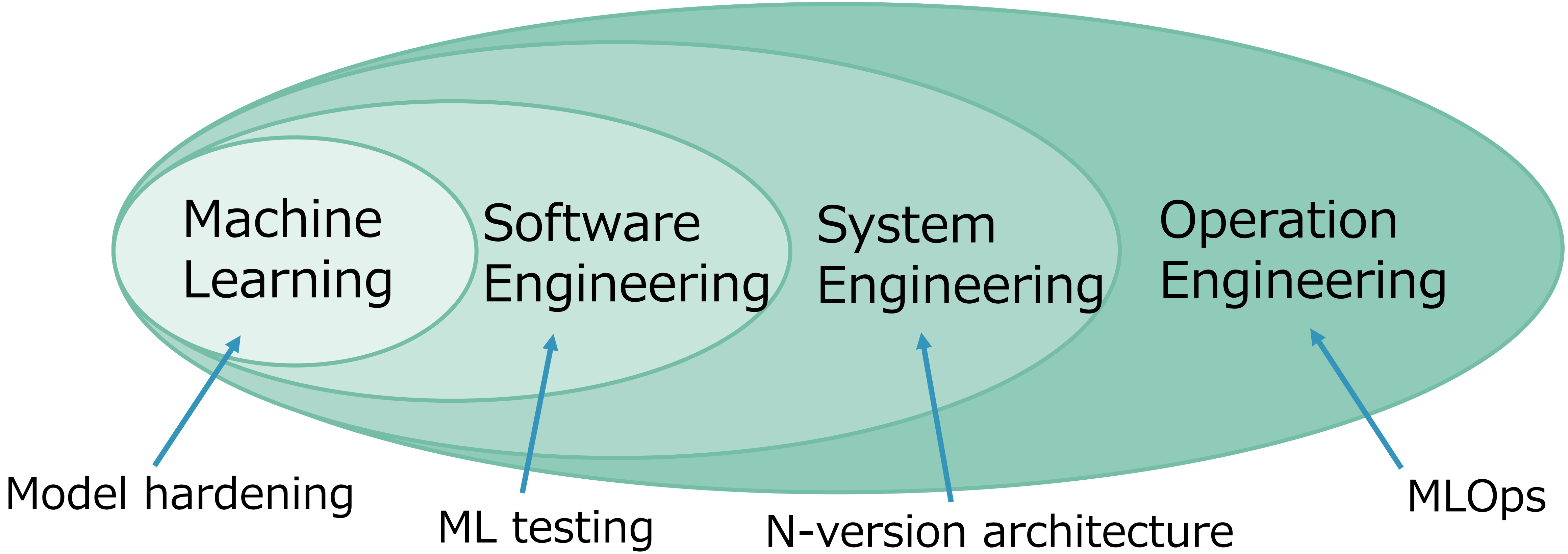
https://shorturl.at/JIob5

GPT-4V often made mistakes when describing the medical image

https://shorturl.at/xfqsh

Engineering for ML system reliability

- Layers of approaches



Outline of this talk

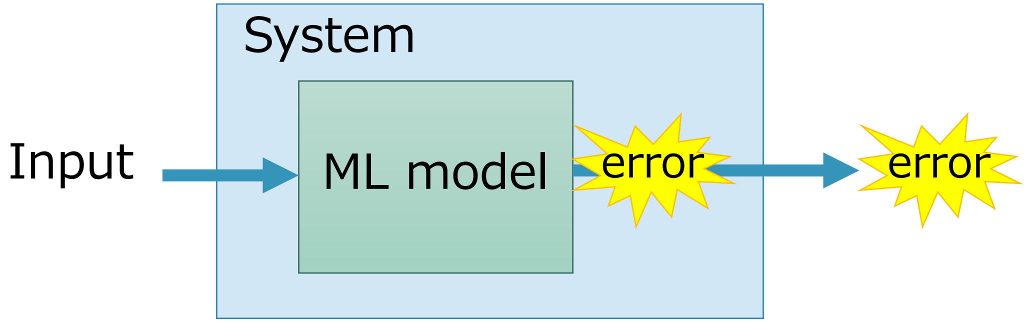
- System engineering
  - **N-version ML architecture** for ML system reliability
- Operation engineering
  - **ML system rejuvenation** for safe autonomous driving
  - **ML model maintenance** for high-availability ML system

N-version ML architecture

N-version ML system

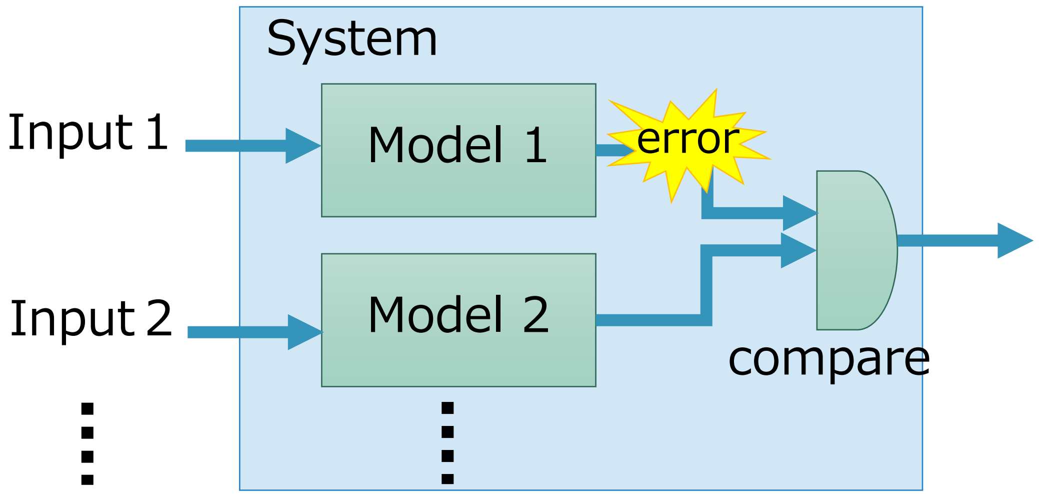
- Suppressing erroneous outputs by multiplexing ML inferences

Relying on a single ML model



Inference errors directly impact the system output

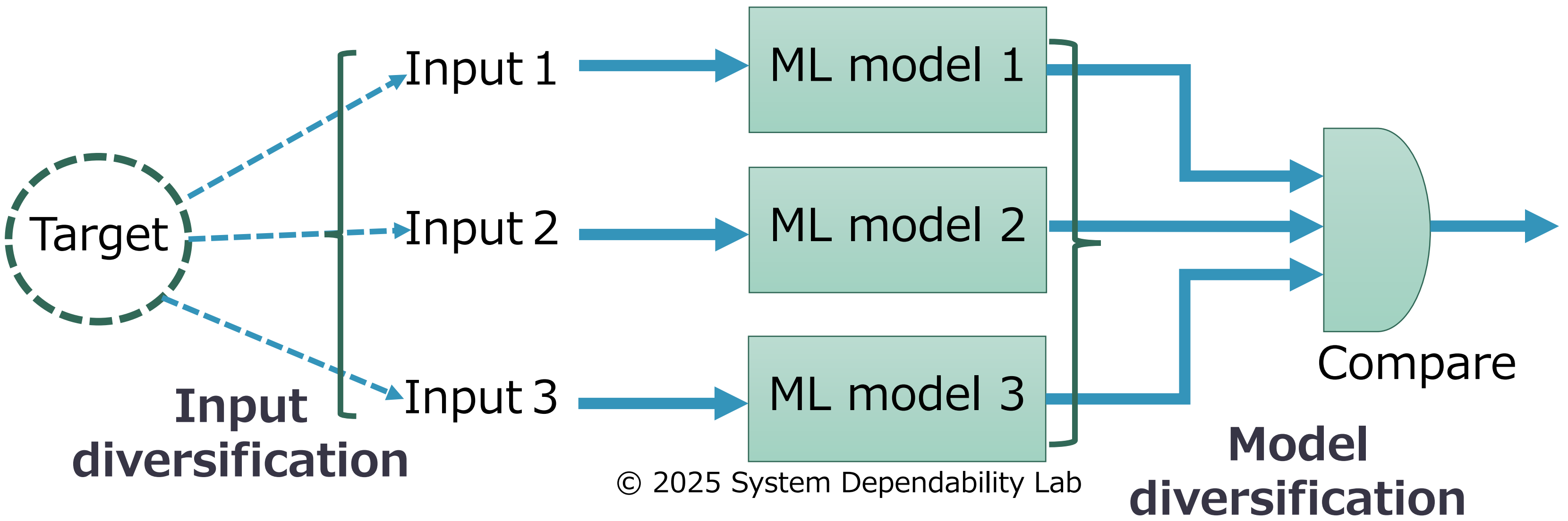
N-version ML system



Inference errors can be detected by comparison

Model diversity & Input diversity

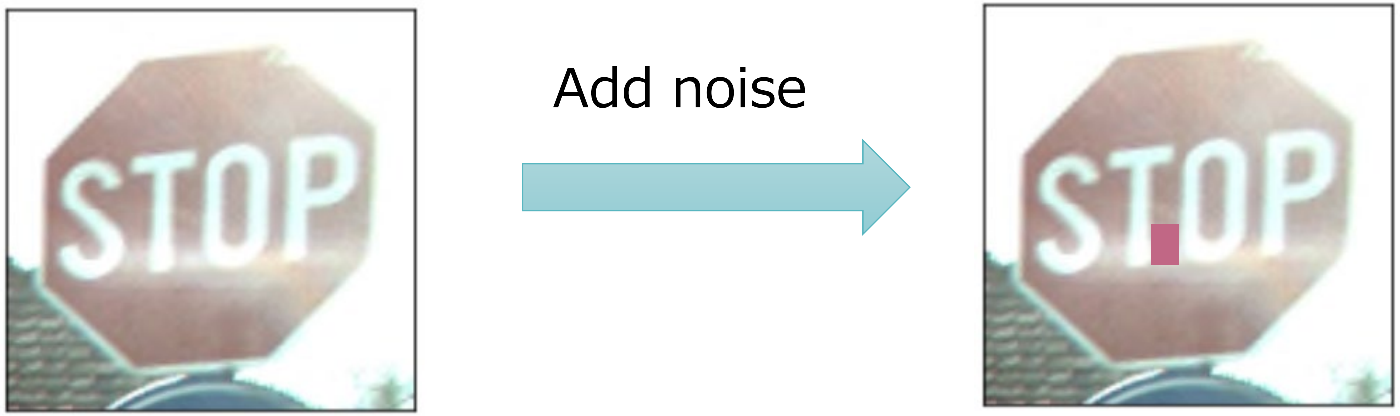
- To diversify multiple ML inferences
- Model diversification
  - Use different ML algorithms and datasets to build ML models
- Input diversification
  - Use different input data sampled from the same target



Input data diversification

- ML models are input sensitive
  - ML models can be fooled by crafted inputs（Adversarial samples)

    → Opposite is also possible



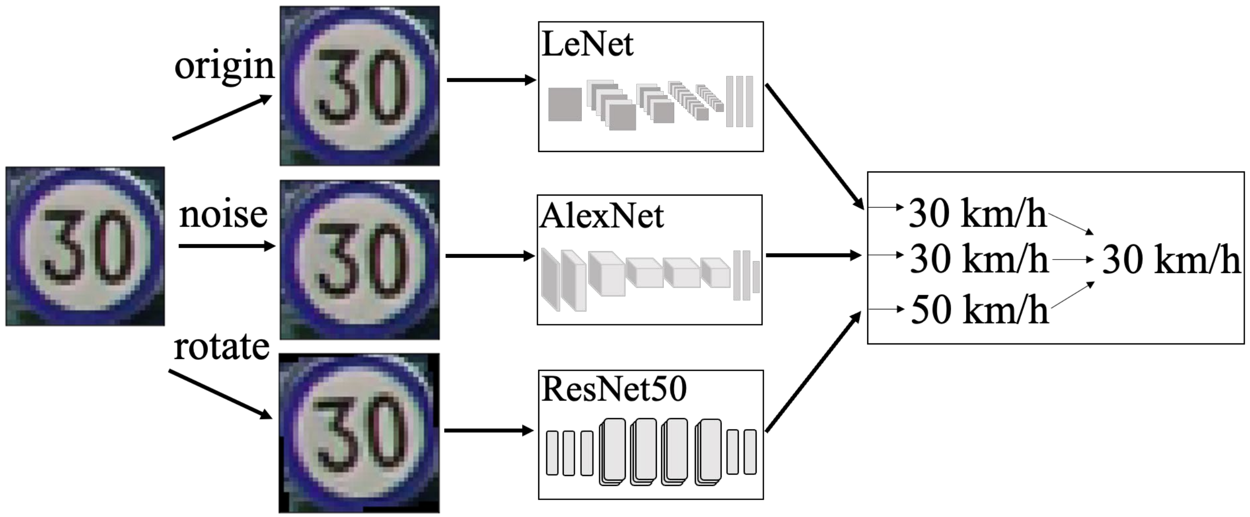
Recognition error **Success!**

Comparison to N-version program

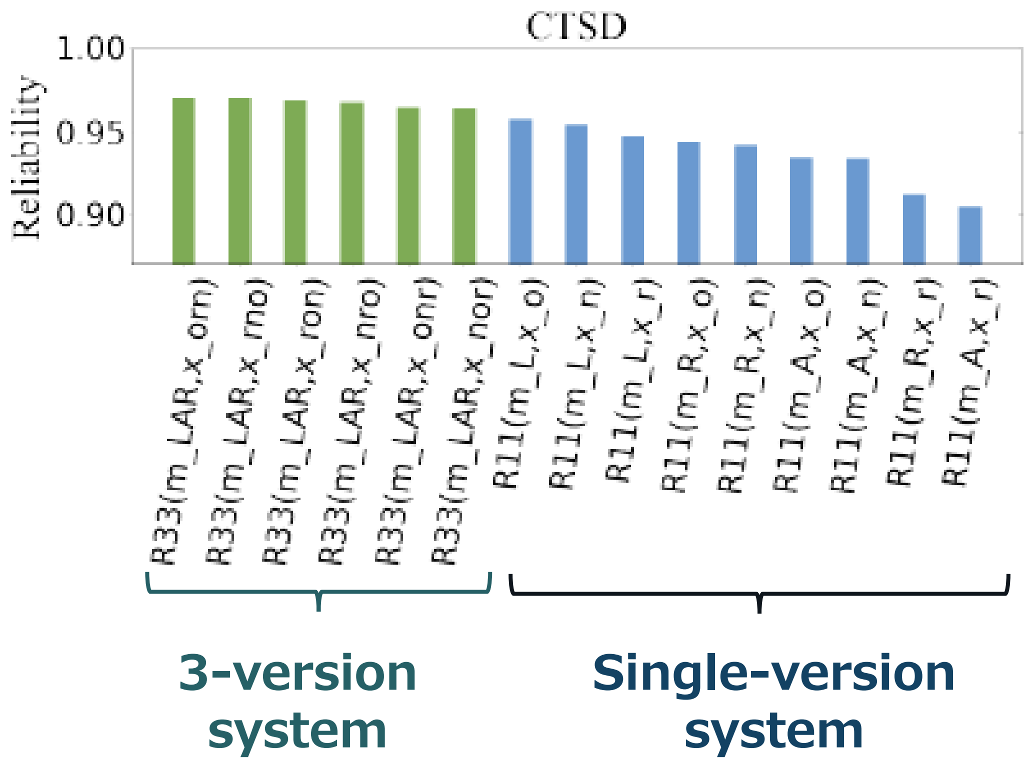
	N-version programming	N-version ML
Target	Software program (generated from specification)	ML module (constructed from data)
Mitigation for	Software faults	Prediction errors
Components to use	Two or more functionally equivalent programs from the same specification	**One or more ML model** for the same task
Sources of diversity	Development teams, programming languages, libraries and tools, etc.	ML algorithms, hyper parameters and **input data**
Cost	high	**Low**

Reliability improvement

- 3-version traffic sign classification systems
  - Consisting of 3 diversified data and 3 deep neural networks

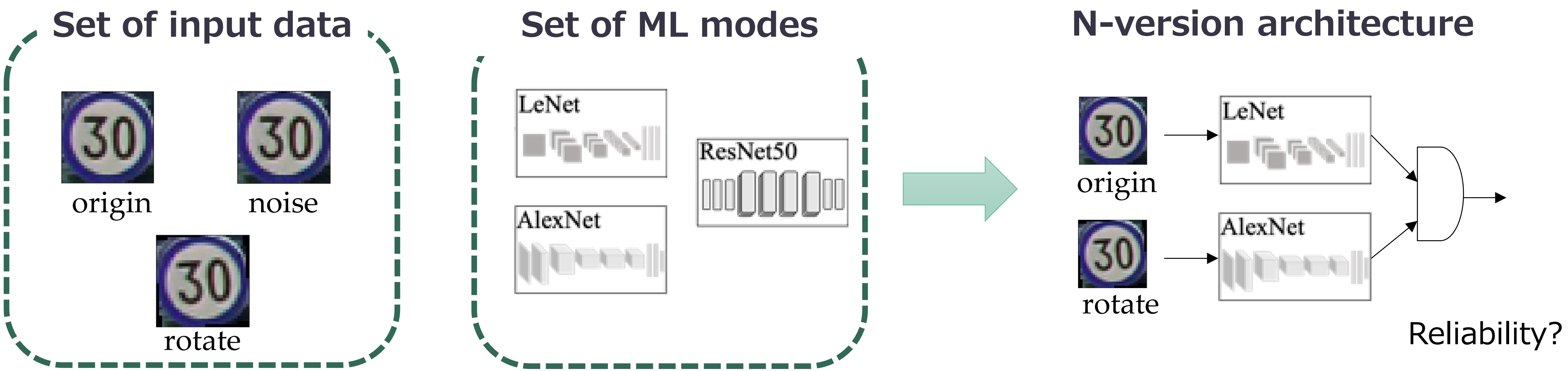


[Q. Wen, et al. ISSRE2023]



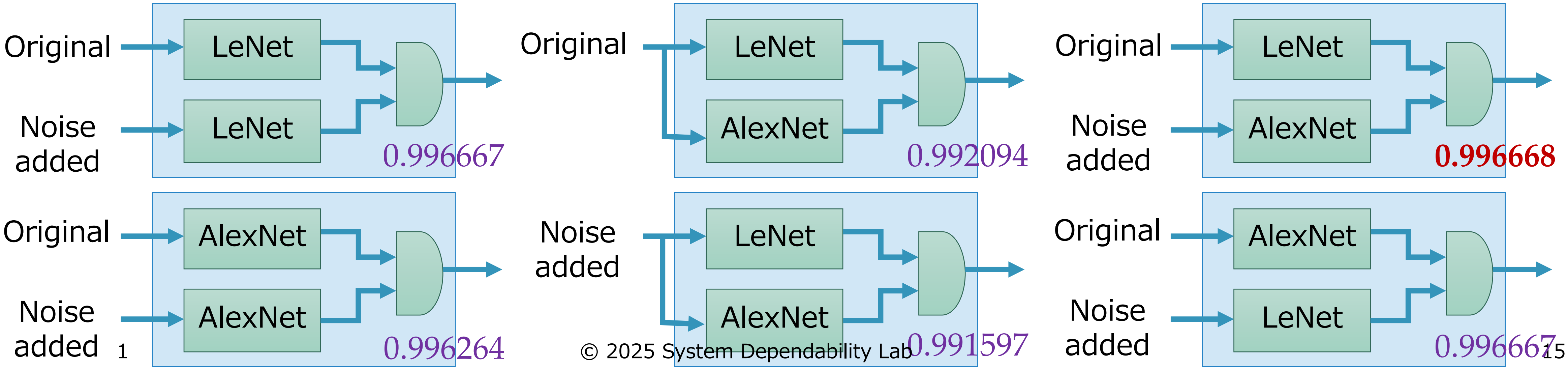
Architecture selection problem

- Given a set of input data and a set of ML models, what is the architecture that can maximize the reliability?
  - Which ML model is used?
  - Which input data is fed to which ML model?



Empirical observation

- Reliability of N-version image classification system depends on the adopted architecture
  - Dataset : MNIST
  - ML models : LeNet, AlexNet
  - Diversified input data : Original, Noise added
  - Decision : Output only when the two versions agree on the results



Reliability model for N-version ML

- Reliability is affected by the combination of input data and ML model

  → Can we theoretically formulate the relation?
- Consider the reliability model for a classification system

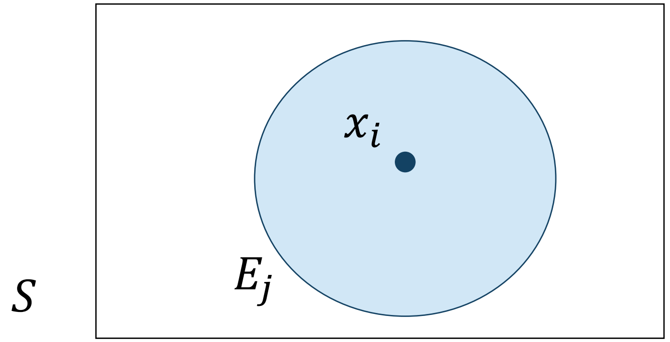
Problem setting

Input data : **Two** input data for the same target
ML model : **Two** ML models for the same classification task
Decision rule : Output only when the two versions agree
Reliability : The probability that the system does not output errors

Reliability of one-version system

- Notation
  - Input data : $x_i, i = \{1,2,\dots\}$
  - ML model : $m_j, j = \{a, b, \dots\}$
  - Sample space of input data : S
  - Error set on which ML model m_j outputs error : $E_j \subset S$
- Reliability of the ML system using m_j for input data x_i

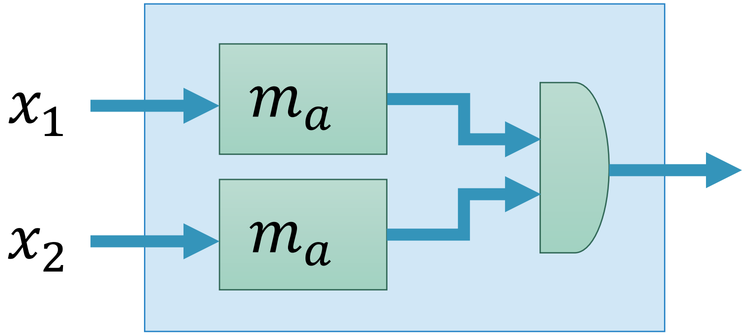
$$1 - \mathrm{P}[x_i \in E_j]$$



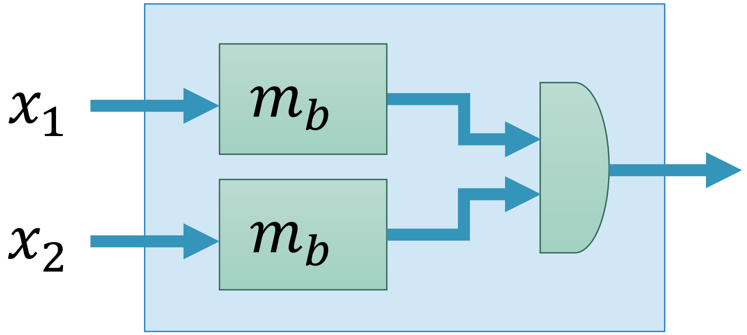
Two-version architectures

6 cases

Single model double input (SMDI)

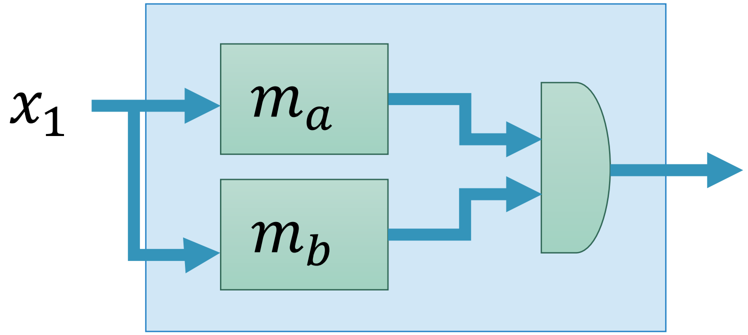


$SMDI(m_a; x_1, x_2)$

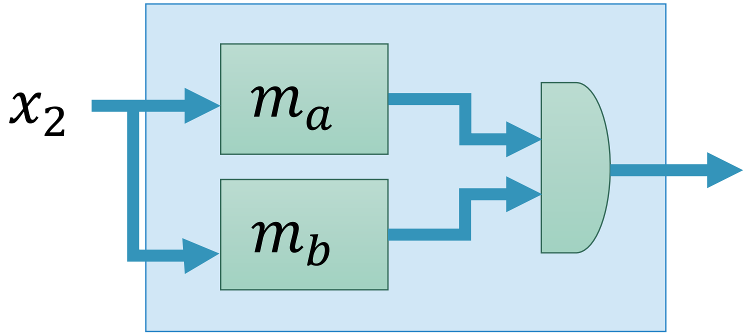


$SMDI(m_b; x_1, x_2)$

Double model single input (DMSI)

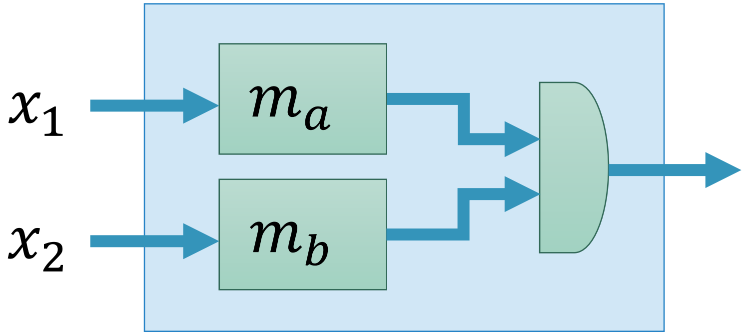


$DMSI(m_a, m_b; x_1)$

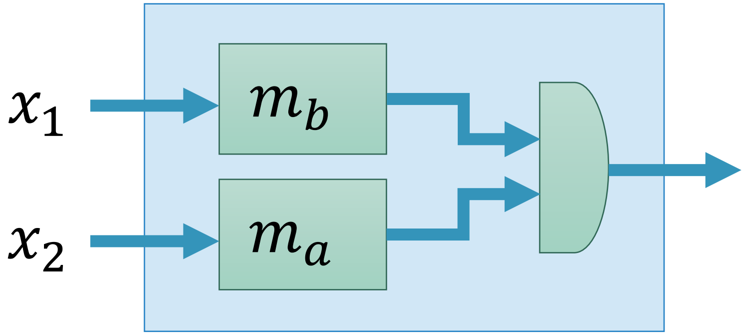


$DMSI(m_a, m_b; x_2)$

Double model double input (DMDI)



$DMDI(m_a; x_1, m_b; x_2)$

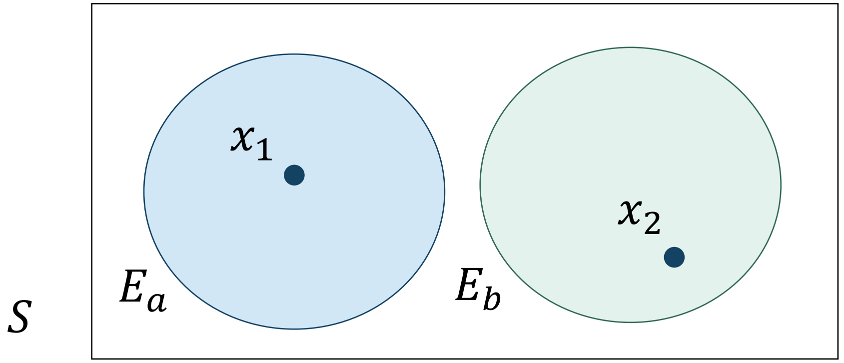


$DMDI(m_a; x_2, m_b; x_1)$

Reliability of Two-version system

- If $\mathrm{P}[x_i \in E_j]$ is independent
  - The error probability of 2-version system is calculated by the product of individual error probabilities

$$1 - \mathrm{P}[x_1 \in E_a] \cdot \mathrm{P}[x_2 \in E_b]$$



- In practice, the independent assumption does not hold
  - Erro set E_j can have intersection
  - Input data x_i does not follow the identical distribution

Diversity metrics

- 2 ML models may have an intersection of error sets

Intersection of errors (Model similarity)

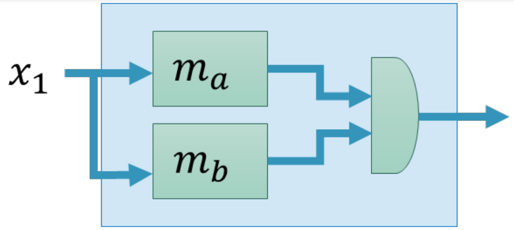
Let E_a, E_b be the subsets of input space S that makes ML models m_a, m_b output errors, respectively. The intersection of errors $\alpha_{b|a,i} \in [0,1]$ is defined by the conditional probability

$$\alpha_{b|a,i} = P[x_i \in E_b | x_i \in E_a] = \frac{P[x_i \in E_a \cap E_b]}{P[x_i \in E_a]}.$$

where $P[x_i \in E_a] > 0$

- Reliability of DMSI system

$$R_{DMSI_{a \cap b,1}} = 1 - \alpha_{b|a,1} \cdot P[x_1 \in E_a]$$



Diversity metrics

- Two input data are not independent

Conjunction of errors (Input similarity)

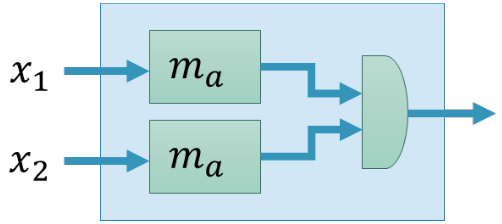
Let x_1, x_2 be the input data for ML model m_j sampled from S. Define conjunction of errors $\beta_{j,2|1} \in [0,1]$ by

$$\beta_{j,2|1} = Pr\left[x_2 \in E_j | x_1 \in E_j\right] = \frac{P\left[x_1 \in E_j, x_2 \in E_j\right]}{P\left[x_1 \in E_j\right]}.$$

where $P\left[x_1 \in E_j\right] > 0$

- Reliability of SMDI system

$$R_{SMDI_{a,1\cap 2}} = 1 - \beta_{a,2|1} \cdot P[x_1 \in E_a]$$

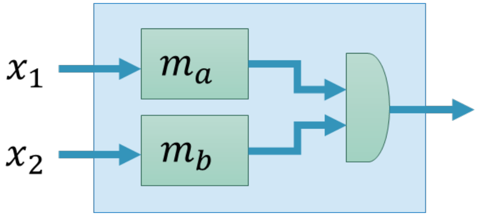


Reliability of DMDI system

- Both model similarity and input data similarity impacts the reliability
- Reliability of $DMDI(m_a; x_1, m_b; x_2)$

$$R_{DMDI_{a,1\cap b,2}} = 1 - \left[\alpha_{b,2|a,1\cap 2} \cdot \beta_{a,2|1} + \alpha_{b,2|a,1\cap \overline{2}} \cdot \left(1 - \beta_{a,2|1}\right)\right] \cdot P[x_1 \in E_a]$$

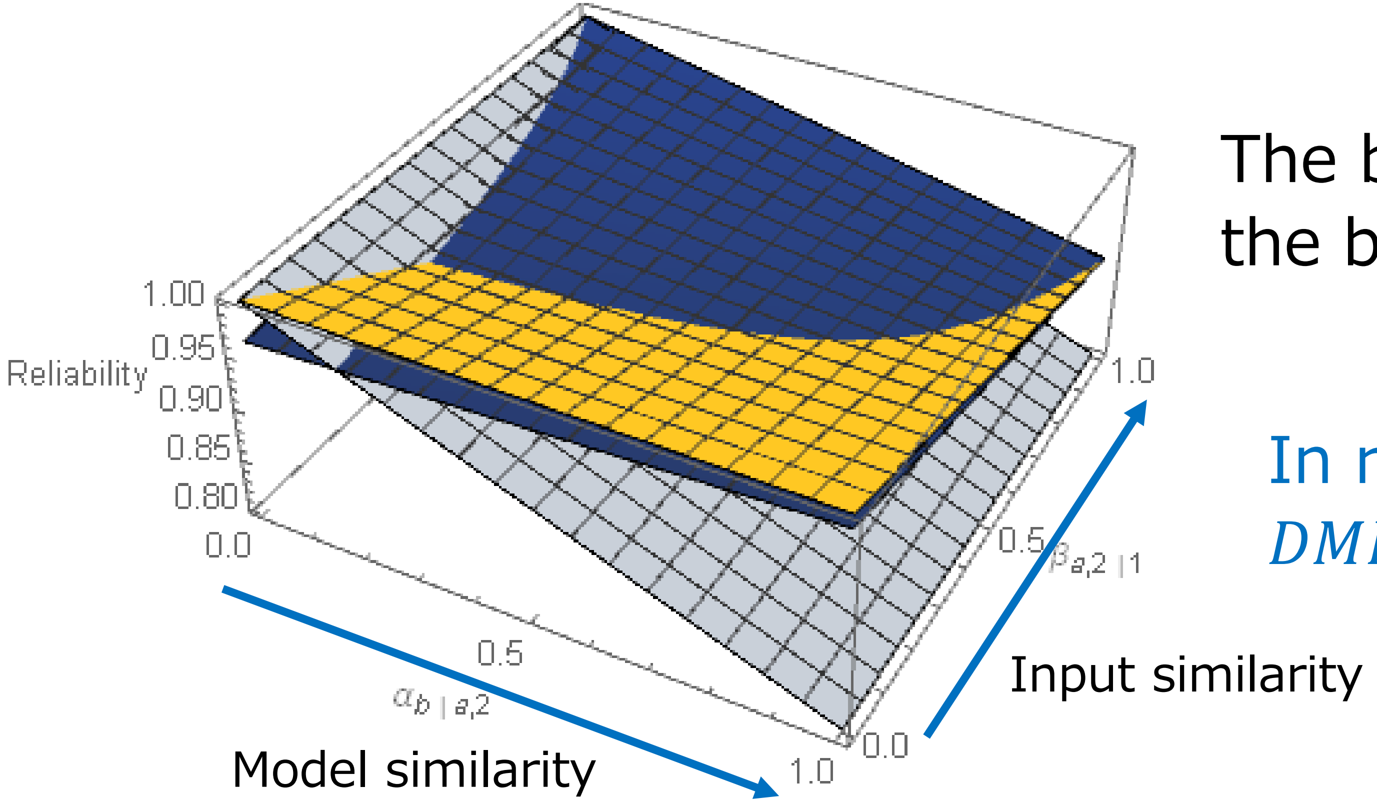
$$\alpha_{b,2|a,1\cap 2} = P[x_2 \in E_b | x_2 \in E_a, x_1 \in E_a]$$
$$\alpha_{b,2|a,1\cap \overline{2}} = P[x_2 \in E_b | x_2 \in \overline{E_a}, x_1 \in E_a]$$



The reliability is characterized the parameters associated with Input similarityとModel similarity

Numerical example

- Under the conditional independence assumption of model similarity and input similarity



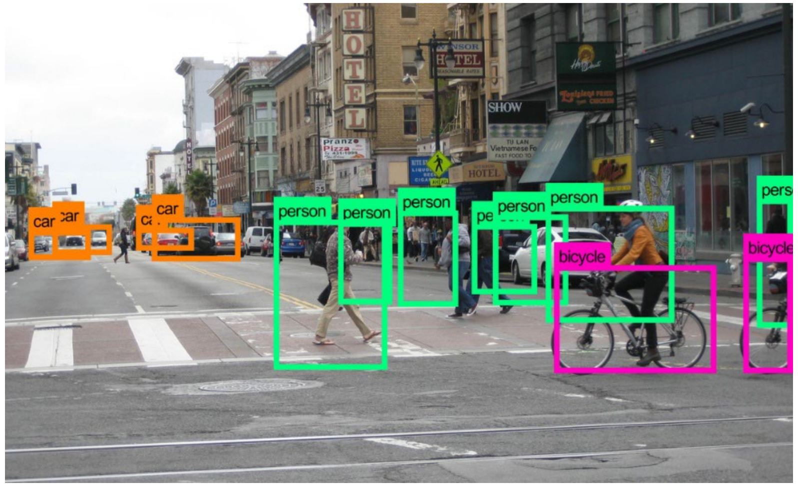
The best architecture is determined by the balance between $\alpha_{b|a,2}$ and $\beta_{a,2|1}$

In realistic scenario in practice, $DMDI_{a,1\cap b,2}$ is preferable architecture

ML system rejuvenation

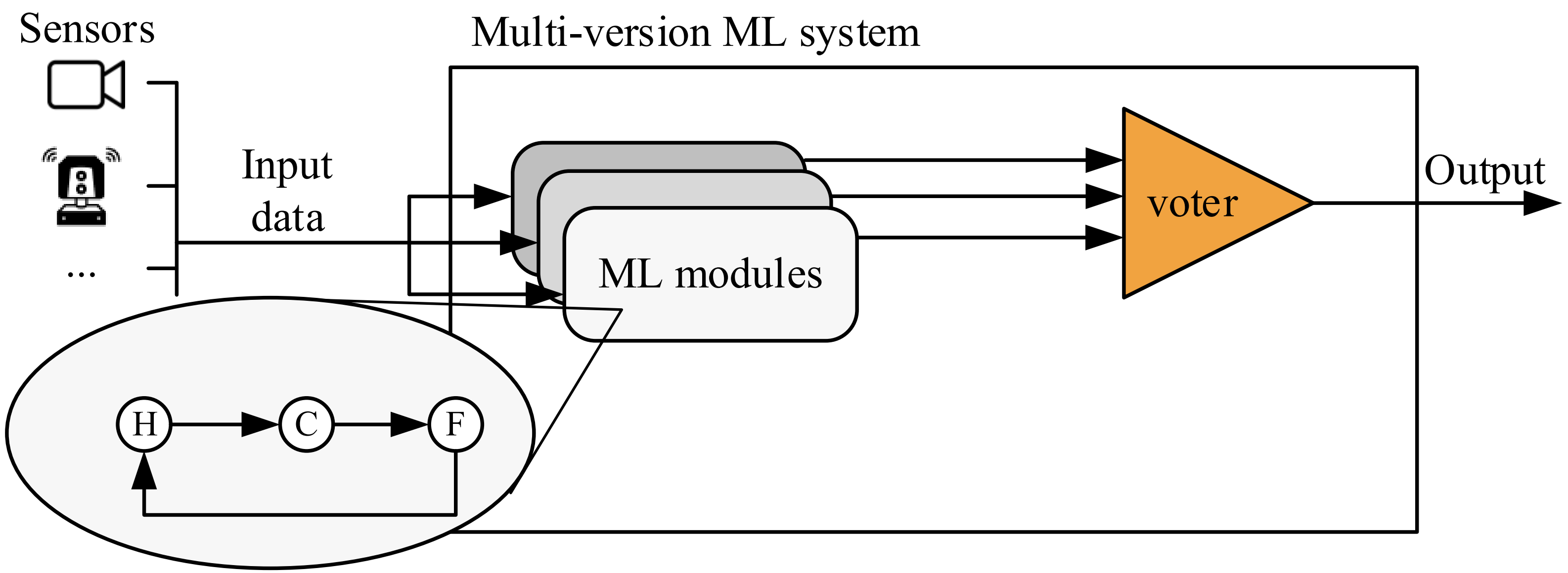
Perception system

- Perception systems are one of the most crucial ML-based components for autonomous vehicles
- Perception systems are also subject to faults and malicious attacks, impacting safety
  - e.g., bit-flip errors and adversarial attacks



N-version perception system

- N-version architecture using multiple object detection models
- Each object detection model degrades gradually
  - Healthy → Compromised (but functional) → Faulty (Non-functional)



AV simulator experiments

- 3-version perception tolerates at most one compromised model
- However, safety is not guaranteed with more severe cases
  → Recovery is needed

System state	YOLO Model	1st collision frame	Total frames	Collision rate%	# Collisions
Three-version					
(3,0,0)	v5s, v5m, v5l	NA	682	0	0/10
(2,1,0)	v5s, v5m, v5m_FI	NA	693	0	0/10
(2,1,0)	v5s, v5m, v5s_FI	NA	682	0	0/10
(1,2,0)	v5s, v5s_FI, v5m_FI	272	666	28.82	5/10
(1,2,0)	v5m, v5s_FI, v5m_FI	335	654	33.08	7/10
(0,3,0)	v5s_FI, v5m_FI, v5l_FI	187	643	57.00	8/10

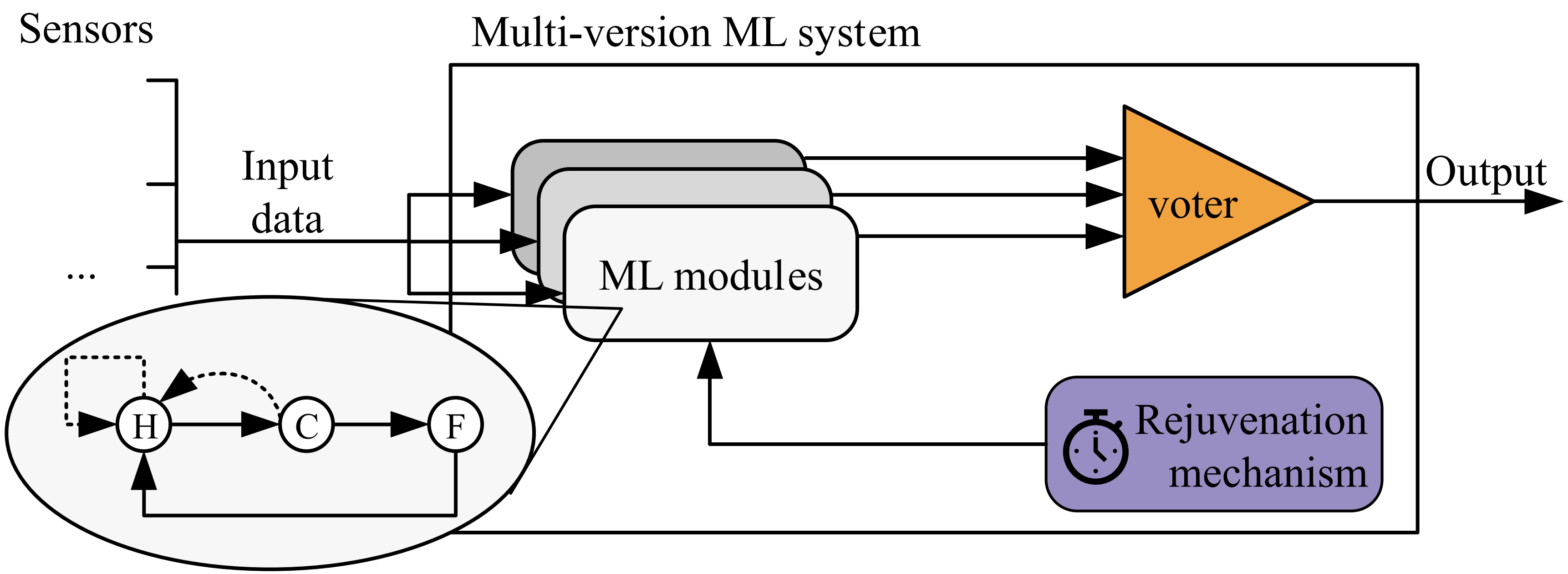
Safe

Unsafe

Number of compromised models

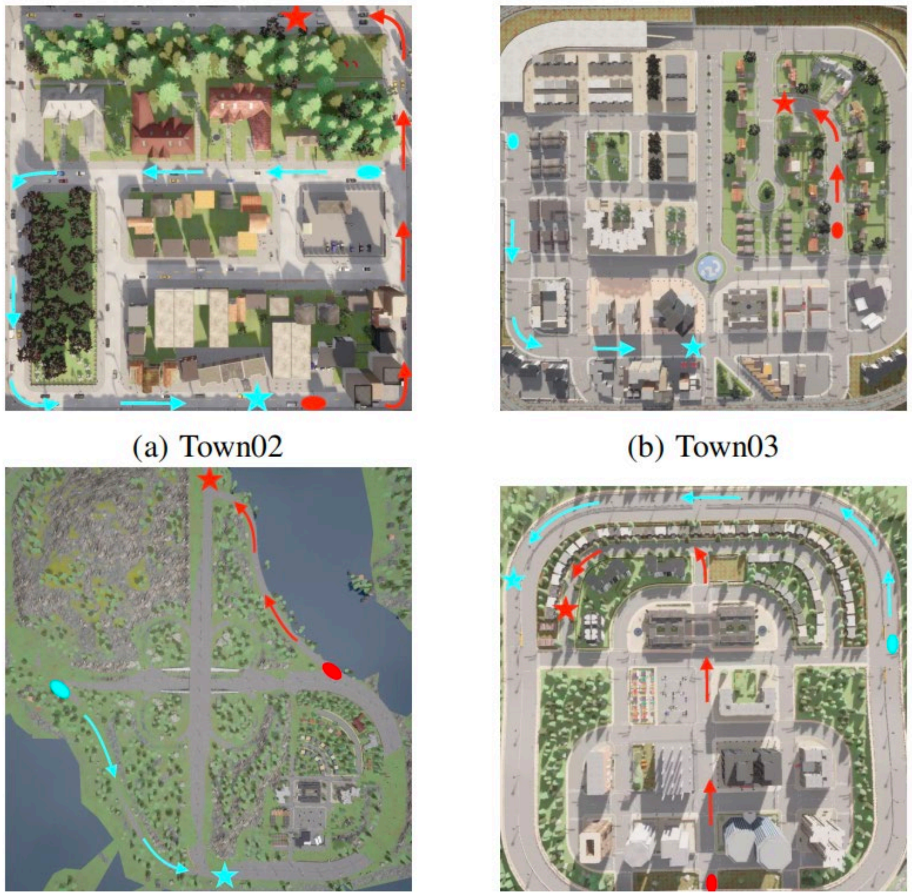
ML model rejuvenation

- Compromised ML models can be rejuvenated periodically to keep safety
  - Deploy a healthy ML model and initialize the ML module



Safety evaluation with AV simulator

- Simulation tools and environment
  - Carla AV simulator
  - Cooperative driving co-simulation framework OpenCDA
- Object detection model
  - YOLOv5s6, YOLOv5m6, YOLOv5l6
- Safety metrics
  - Collision rate
  - First collision frame number

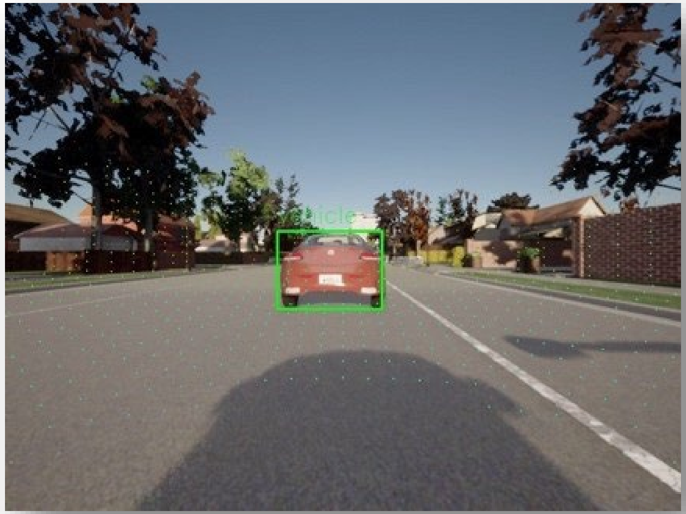


(a) Town02 (b) Town03 (c) Town04 (d) Town05

Fault injection experiments

- Compromised versions of YOLOv5 models
  - Use PyTorchFI to change YOLOv5’ parameters randomly
  - Compromised detection model fails to detect the vehicle, resulting in a collision

Healthy model



Compromised model

Evaluation results

- The system with rejuvenation achieves 0% collision rates across all tested routes

Route	1st coll.		Total frames		Coll. rate (%)		#Coll.	
	w/	*w/o*	*w/*	*w/o*	*w/*	*w/o*	*w/*	*w/o*
#1	NA	299	610	618	0.00	9.70	0/5	4/5
#2	NA	268	735	675	0.00	12.89	0/5	3/5
#3	NA	203	630	543	0.00	47.98	0/5	4/5
#4	NA	390	720	730	0.00	42.45	0/5	4/5
#5	NA	313	644	757	0.00	52.25	0/5	5/5
#6	NA	383	663	684	0.00	33.97	0/5	4/5
#7	NA	204	626	661	0.00	14.91	0/5	4/5
#8	NA	241	630	680	0.00	54.13	0/5	5/5
Avg/*Total*	NA	287	657	669	0.00	33.54	*0/40*	*33/40*

Rejuvenation interval

- The shorter intervals enhance driving safety by quickly recovering compromised models

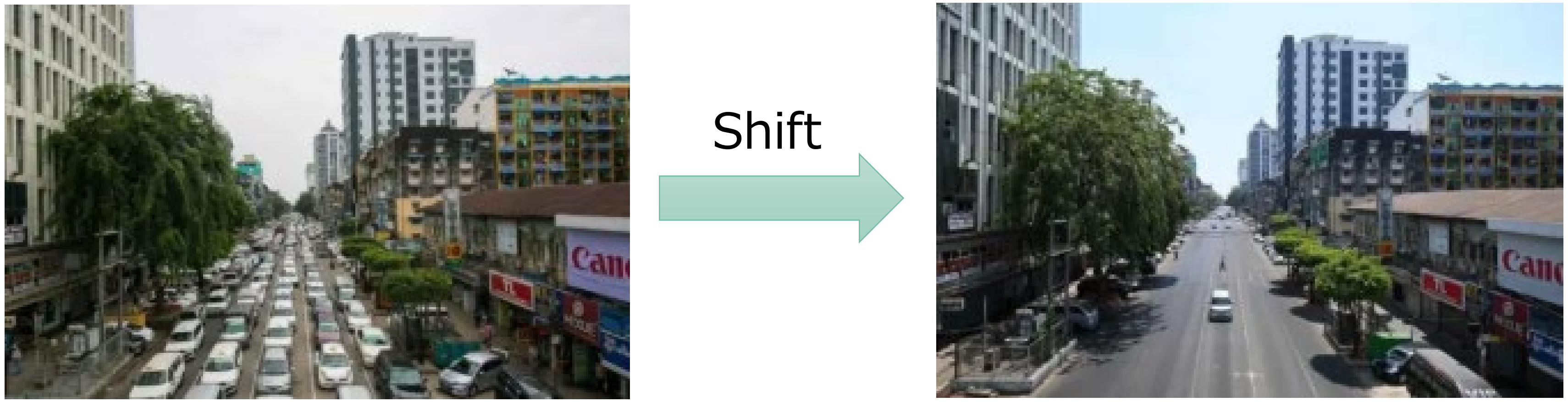
Rejuvenation interval

$1/\gamma$ (s)	1st coll.	Total	Coll. rate	#Coll.
3	NA	610	0.00%	0/5
5	526	627	1.27%	1/5
7	246	574	8.93%	2/5
9	270	632	10.44%	3/5
Avg/*Total*	347	611	5.16%	*6/20*

ML model maintenance

Dataset shift

- The performance of ML models deteriorates when input data distribution changes
  - Sample selection bias
  - Non-stationary environment
- Model retraining is essential to maintain long-term performance



https://www.bbc.com/news/world-asia-52677139

Model retraining strategies

- Availability of the ML system is affected by the frequency of retraining attempts
- Progressive retraining policy
  - ML models are constantly retrained with new data
- Conservative retraining policy
  - ML models are retrained when observing performance failure

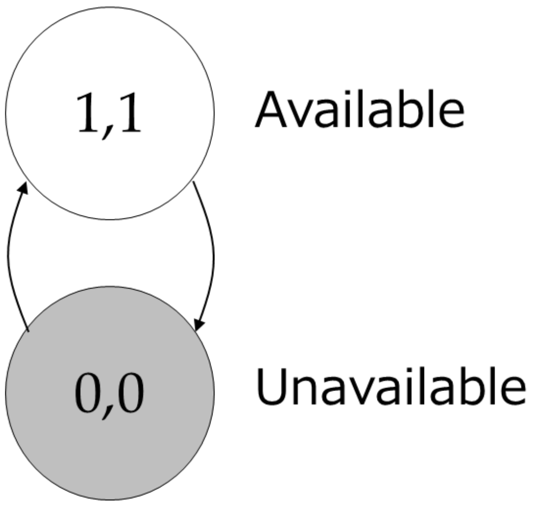
How can we compare the effectiveness of these retraining policies?

Availability modeling

- Two-component ML system
- ML system is available when the performance of the downstream model satisfies threshold τ_d
- Formulate a Continuous-time Markov Chain (CTMC)
  - System state (u, d)

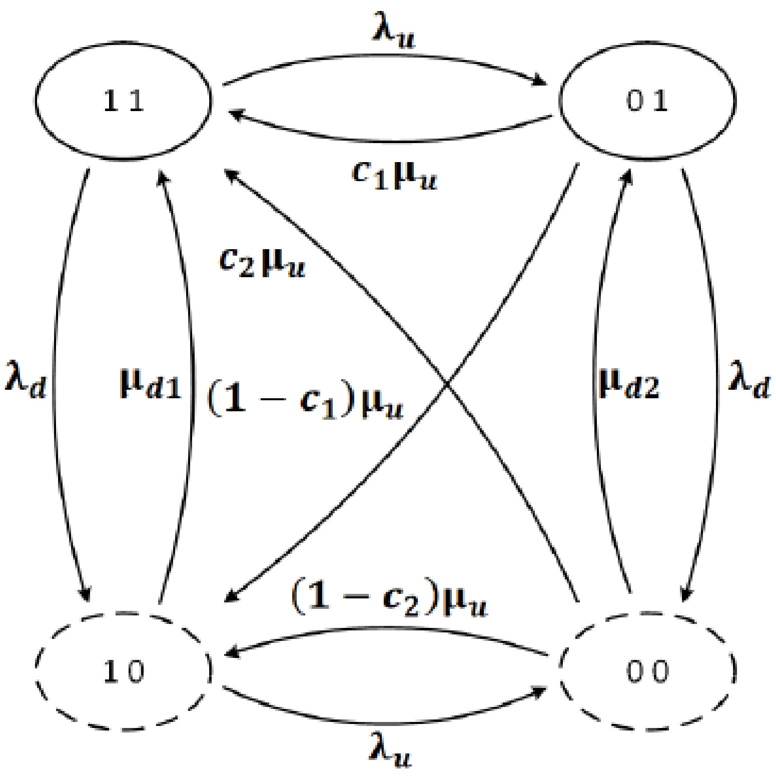
Upstream model → u; Downstream model → d

$$u, d = \begin{cases} 1, & \text{Satisfying the threshold} \\ 0, & \text{Unacceptable} \end{cases}$$

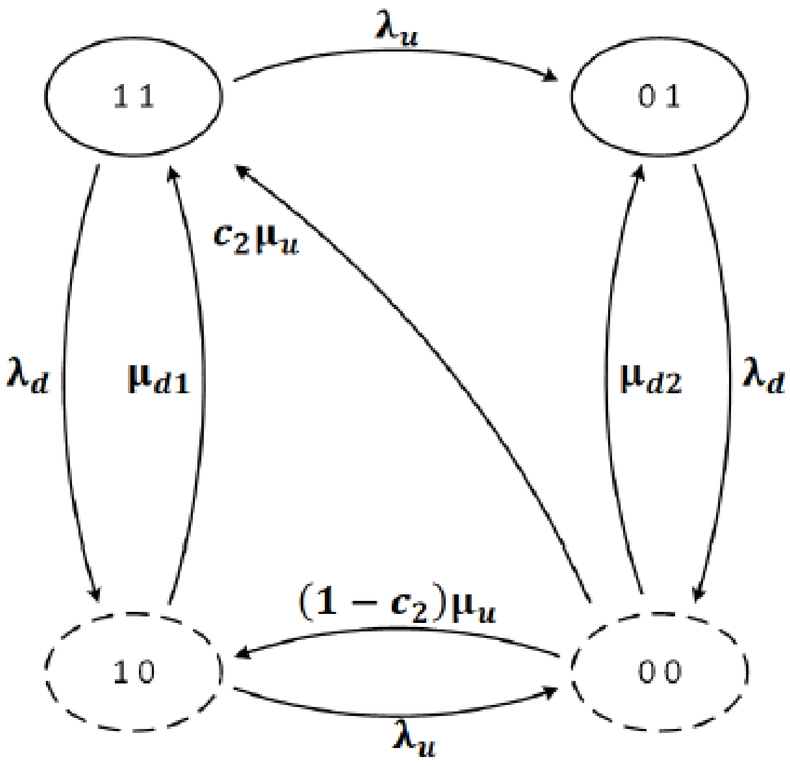


CTMCs

Progressive
retraining policy



Conservative
retraining policy



Policy comparison

- Each policy has a distinctive advantage over the parameter space

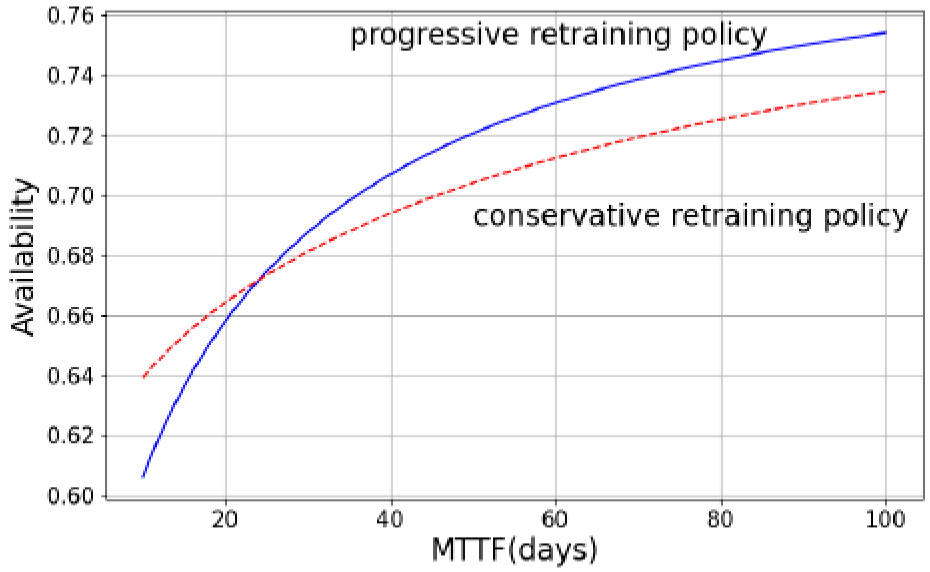


Figure: MTTF for Upstream model

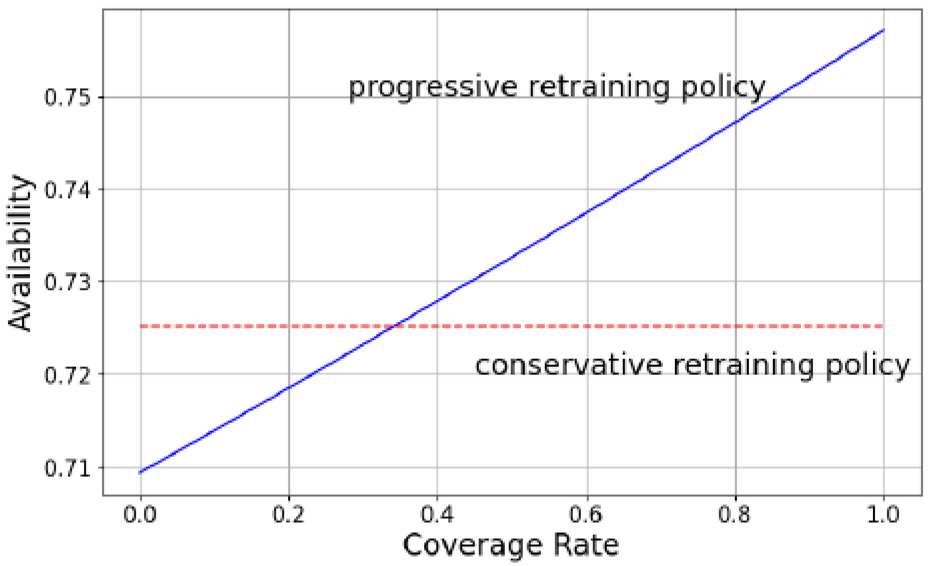


Figure: Coverage Factor c_1

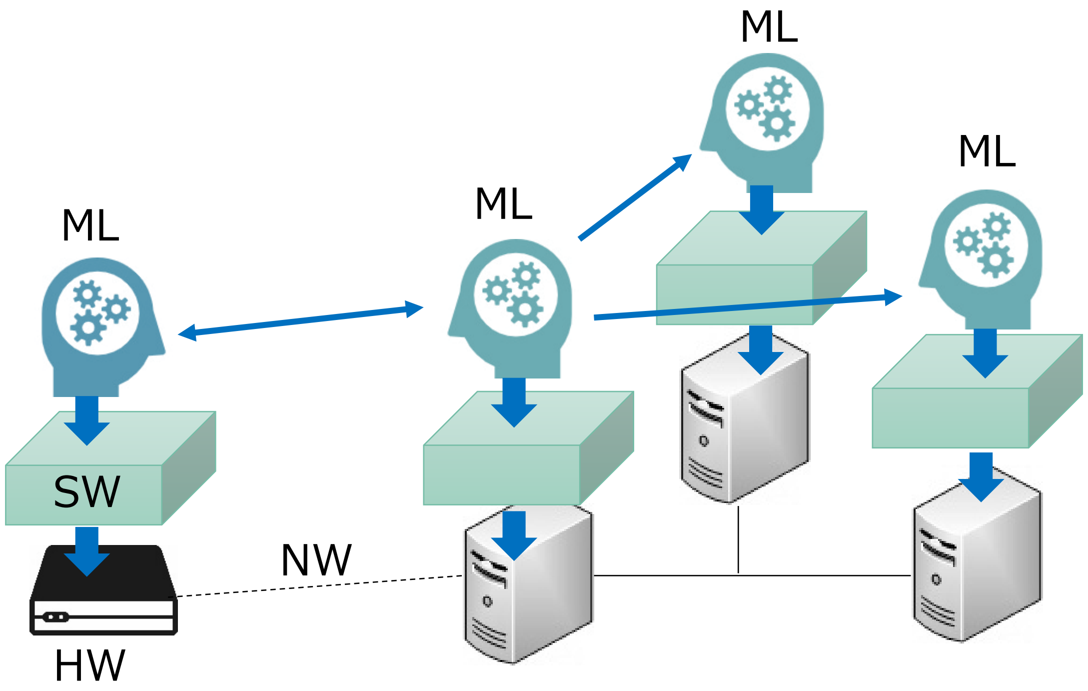
Future challenges

Reliability of generative AI systems

- Quantitative reliability evaluation for AI systems involving ML models for generative tasks
- System and operation engineering for reliable generative AI systems

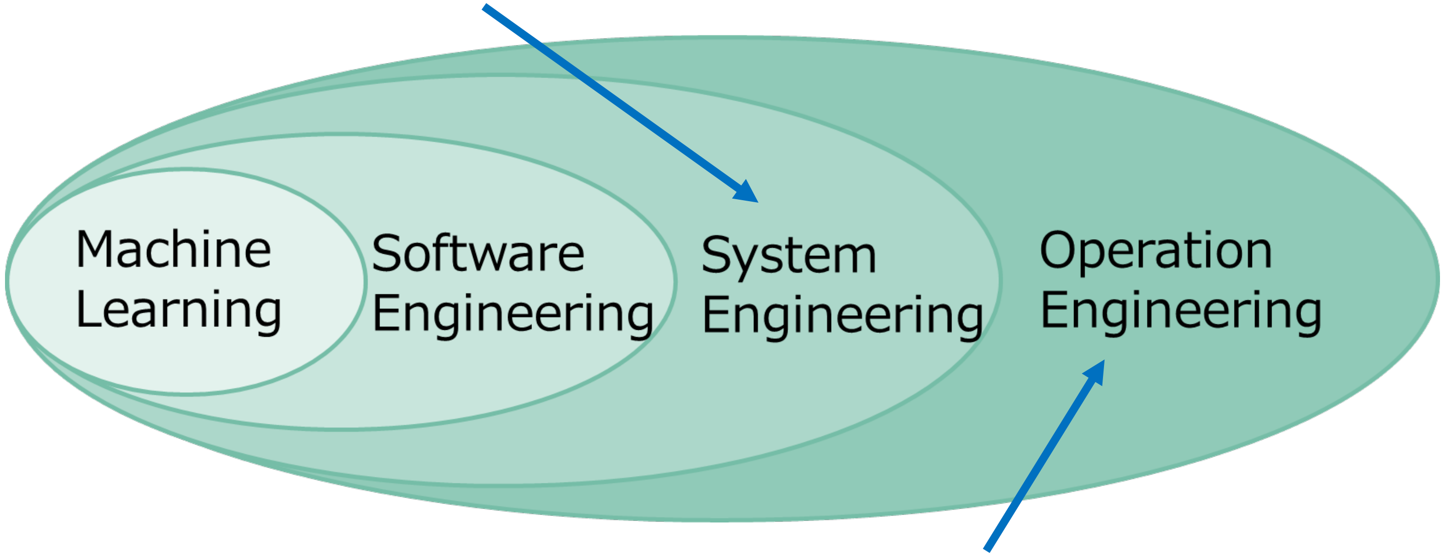
High availability ML systems

- ML systems' reliability needs to be maintained for long term
- Degradation issues
  - Model aging
  - Software aging
- Dependency issues
  - Between multiple ML components
  - Reliance on software and hardware



Conclusion

- **N-version ML architecture** for ML system reliability



- **ML system rejuvenation** for safe autonomous driving
- **ML model maintenance** for high-availability ML systems

Thanks to collaborators

Qiang Wen
(University of Tsukuba)

Zhengji Wang
(Rakuten Bank)

Júlio Mendonça
(Tilburg University)

Marcus Völp
(University of Luxembourg)

Thank you