University of Tsukuba

On the diversity of machine learning models for system reliability

Fumio Machida
University of Tsukuba

3rd December, 2019

In 24th IEEE Pacific Rim International Symposium
on Dependable Computing (PRDC 2019)

Outline

1. **Quality issue of Machine Learning (ML) systems**
2. Diversity of ML models
3. Experimental study
4. System reliability model and analysis
5. Related work
6. Conclusion

ML application systems

ML is an important building block of intelligent software systems

- ML applications

Autonomous vehicle

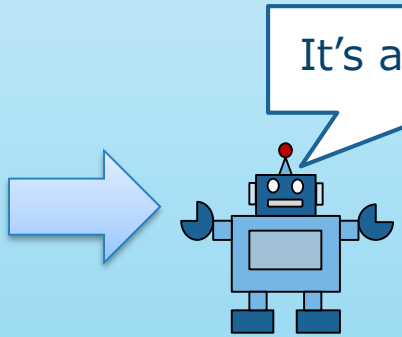
Voice assistant device

Factory automation robot

Reliability concern in ML systems

Uncertain outputs of ML components cause the unreliability of the system

- Outputs of ML model are uncertain
  - Functional behavior is determined by training data



99%
accurate!!

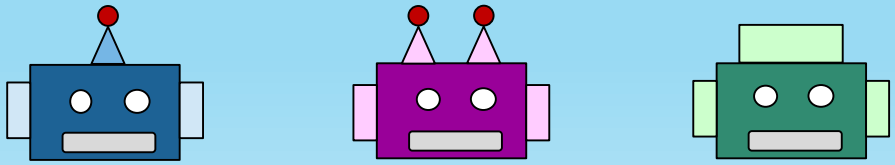
… but what if 1% happens

System reliability design is crucial

Toward reliable ML systems

Diversity of outputs from ML modules can be a clue to improve system reliability

- Idea
  - Applying "N-version programming" to ML systems
    - Under N-version programming system, even when one software component outputs an error, another version can mask the error
  - Increasing the diversity of ML modules' outputs so that each module makes errors independently



Research questions

RQ1

- **How can we diversify the outputs from different ML models for the same task?**

RQ2

- **How can we use the diverse ML models to improve the system reliability?**

Outline

1. Quality issue of Machine Learning (ML) systems
2. **Diversity of ML models**
3. Experimental study
4. System reliability model and analysis
5. Related work
6. Conclusion

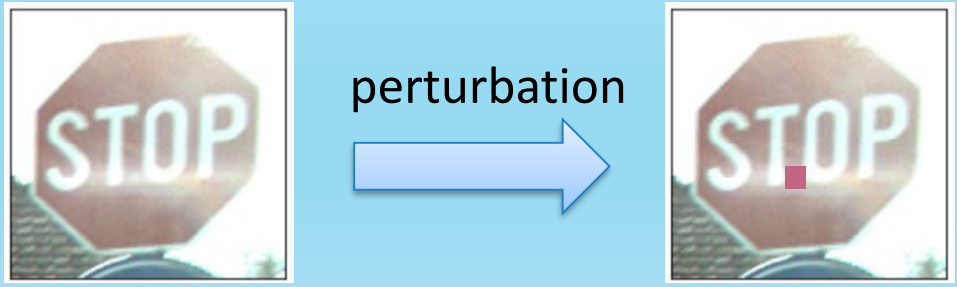
Diversity of ML models

- Potential contributing factors to improve the diversity of ML modules
  - Training data
  - ML algorithm
    - hyper-parameter
    - network architecture
  - Input data for prediction

Input data for prediction

We can diversify the output of ML modules by varying input data in the operation

- Sensitivity to input data
  - A subtle perturbation of input data can easily fool a ML model to output error (Adversarial example)
  - Opposite can also happen. Just a subtle perturbation of input data can flip an error case to a correct output



Error case **Success case**

Outline

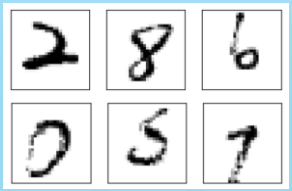
1. Quality issue of Machine Learning (ML) systems
2. Diversity of ML models
3. **Experimental study**
4. System reliability model and analysis
5. Related work
6. Conclusion

Experimental study

To address RQ1, we investigated the outputs of diverse ML models and inputs

- Objective
  - Not on the benchmark of different ML models
  - But on characterizing the difference of error spaces of input data by various ML models

Data sets



MNIST handwritten digit

Belgian Traffic Sign

ML algorithms

- Random forest (RN)
- Support vector Machine (SVM)
- Convolutional neural networks (CNN)

Diversity affected metric

Coverage of errors are defined to quantify the benefits from diversity

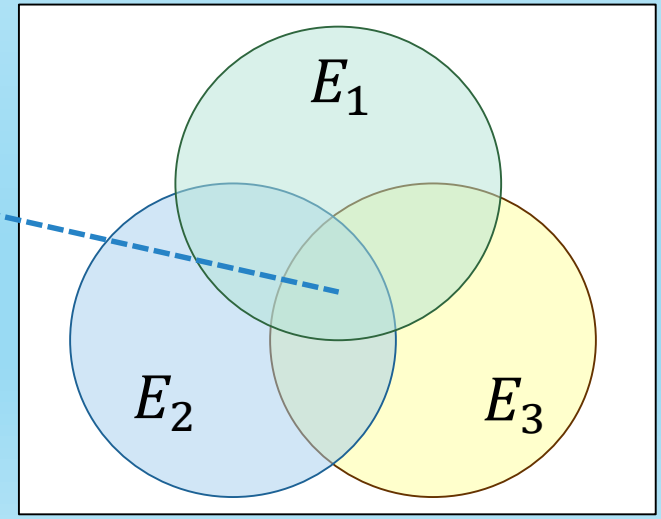
- Error space E_i
  - The subset of sample space for individual ML models that can cause classification errors

- Coverage of errors

$$Cov(\mathcal{M}) = 1 - \frac{\left|\bigcap_{m_i \in \mathcal{M}} E_i\right|}{|S|}$$

$\mathcal{M}$: Set of ML models

Sample space



Algorithm diversity

Using three different ML algorithms to predict the labels of digits

- **RF**
  - The best performed parameters are chosen by a grid search in scikit-learn
- **SVM**
  - Support vector classifier implemented in scikit-learn is used
- **CNN**
  - The network with a convolutional layer, a max pooling layer and a fully-connected layer is configured by Keras

Number of classification errors

CNN achieves the smallest classification errors for all the digits

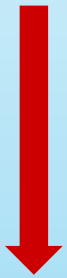
Label	0	1	2	3	4	5	6	7	8	9	Total
$\|S\|$	980	1135	1032	1010	982	892	958	1028	974	1009	10000
$\|E_{CNN}\|$	3	6	11	3	5	9	22	11	11	28	109
$\|E_{RF}\|$	10	13	36	34	26	30	19	37	41	47	293
$\|E_{SVM}\|$	11	12	26	27	32	42	25	39	40	42	296

How the coverage of errors can be improved by adding the other prediction results?

Increased coverage of errors

The coverage of errors is increased by adding the other prediction results

$Cov(CNN)$	0.9891
$Cov(CNN, RF)$	0.9918
$Cov(CNN, RF, SVC)$	0.9934

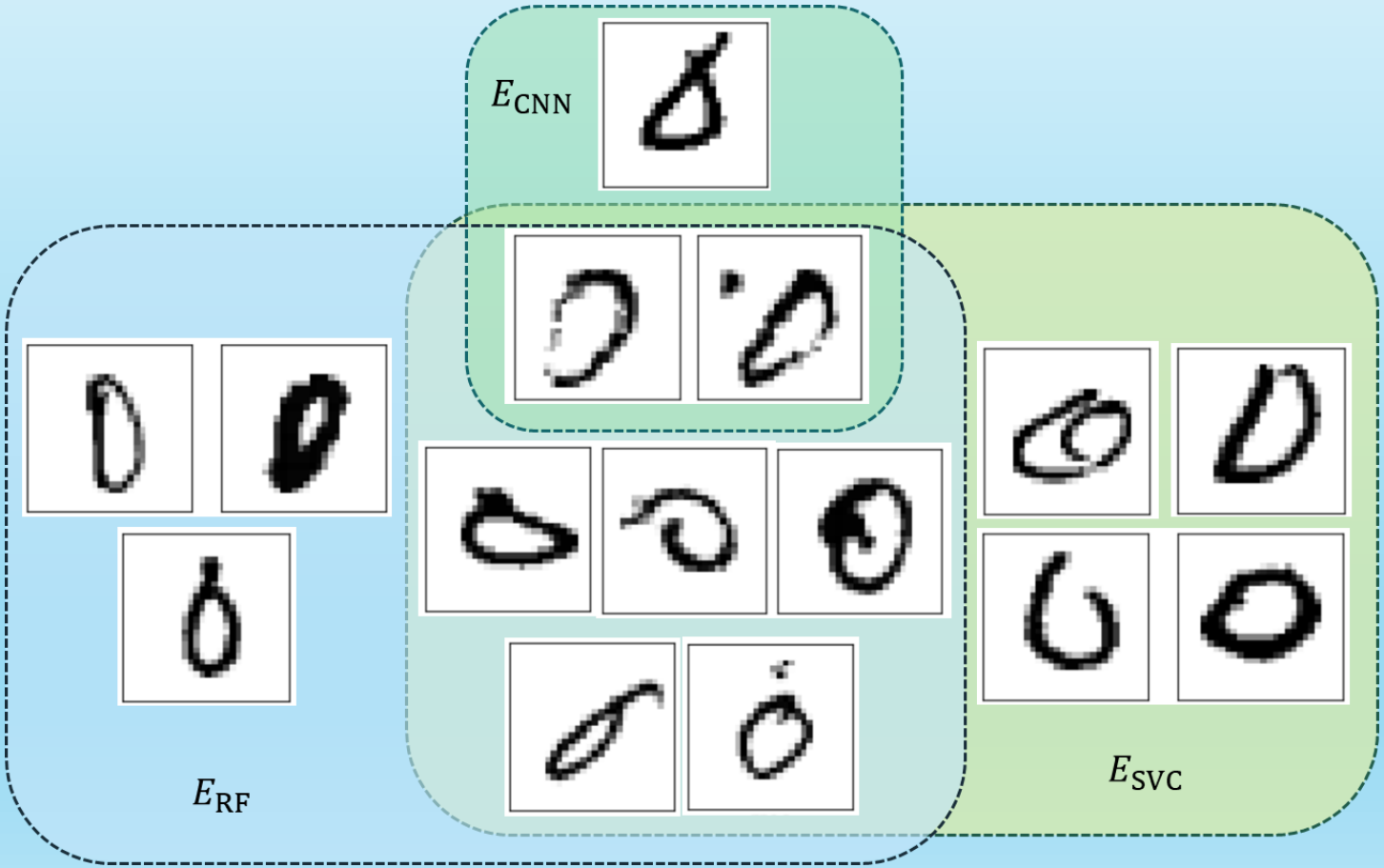


increase

Note that the certainty of accurate prediction is decreased as a result of additional predictions from the other models

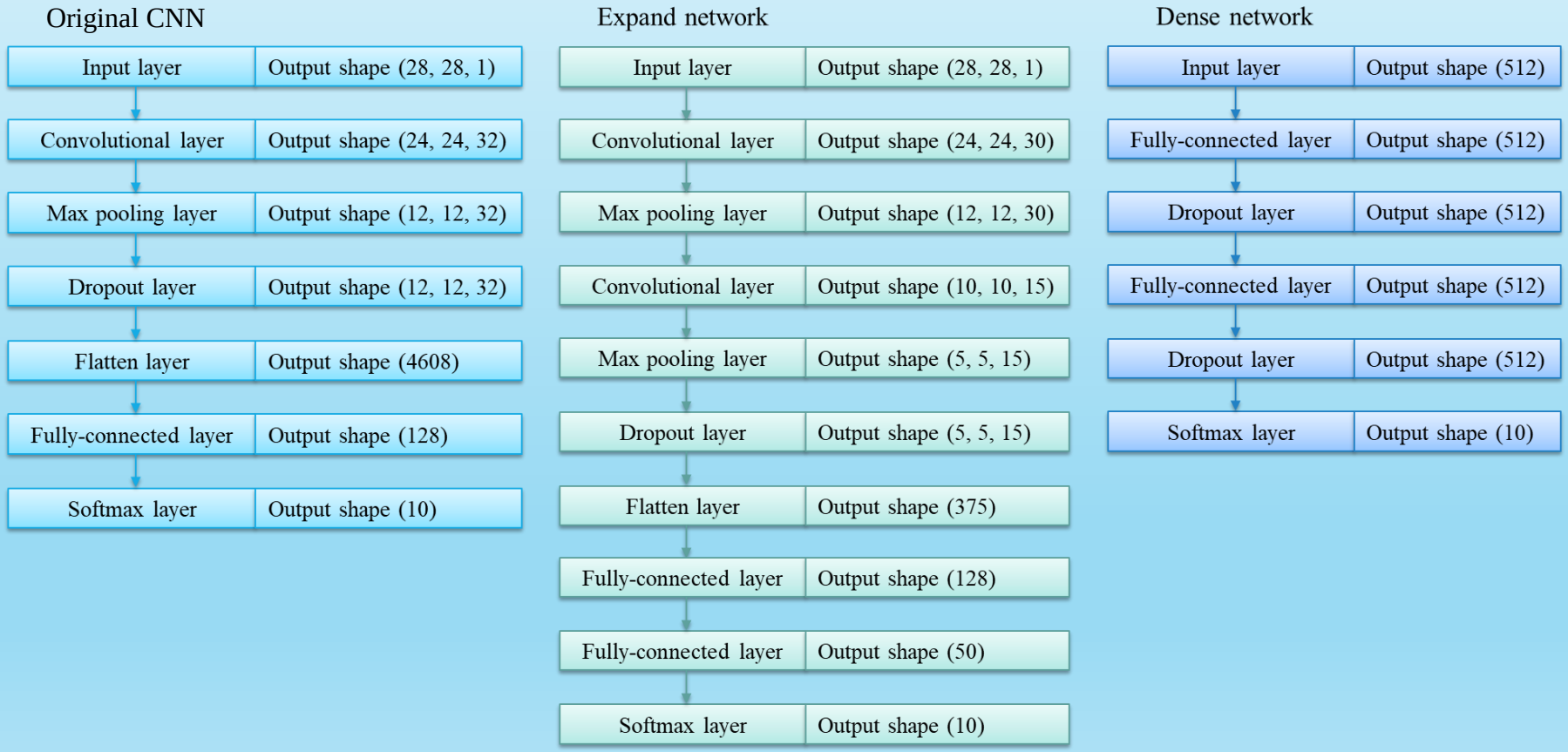
Visualization of error spaces for "0"

- Only two out of 980 samples are not accurately classified by any models ($|E_{\mathrm{CNN}} \cap E_{\mathrm{RF}} \cap E_{\mathrm{SVC}}| = 2$)



Architecture diversity

Using three different neural network architectures to predict the labels of digits



Number of classification errors

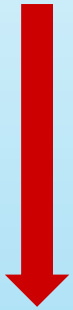
Both of CNN and Expand network achieve good classification accuracy

Label	0	1	2	3	4	5	6	7	8	9	Total
$\lvert S \rvert$	980	1135	1032	1010	982	892	958	1028	974	1009	10000
$\lvert E_{\mathrm{CNN}} \rvert$	3	6	11	3	5	9	22	11	11	28	109
$\lvert E_{\mathrm{Dense}} \rvert$	9	6	12	13	21	19	11	19	22	23	155
$\lvert E_{\mathrm{Expand}} \rvert$	2	9	4	8	12	9	16	11	7	11	89

Increased coverage of errors

The coverage of errors is increased by adding the other neural networks' results

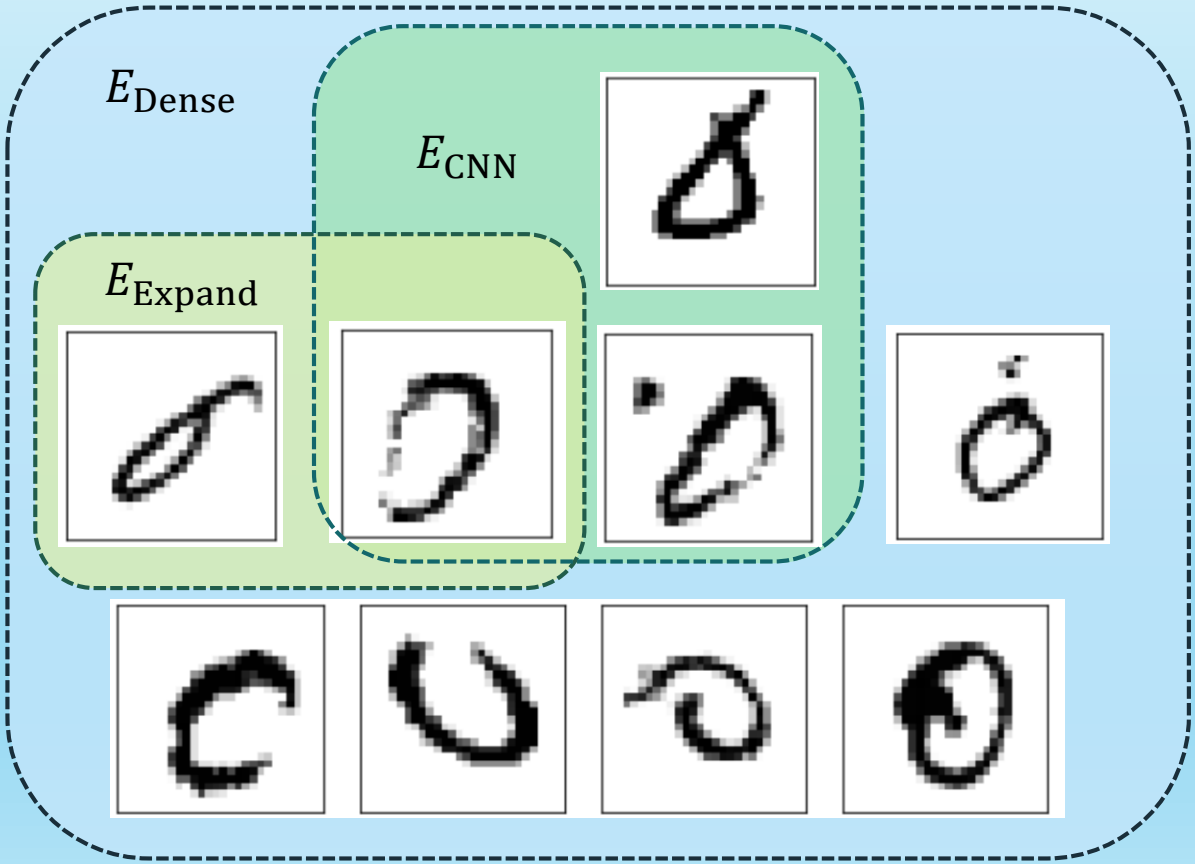
$Cov(CNN)$	0.9891
$Cov(CNN, Dense)$	0.9944
$Cov(CNN, Dense, Expand)$	0.9971



increase

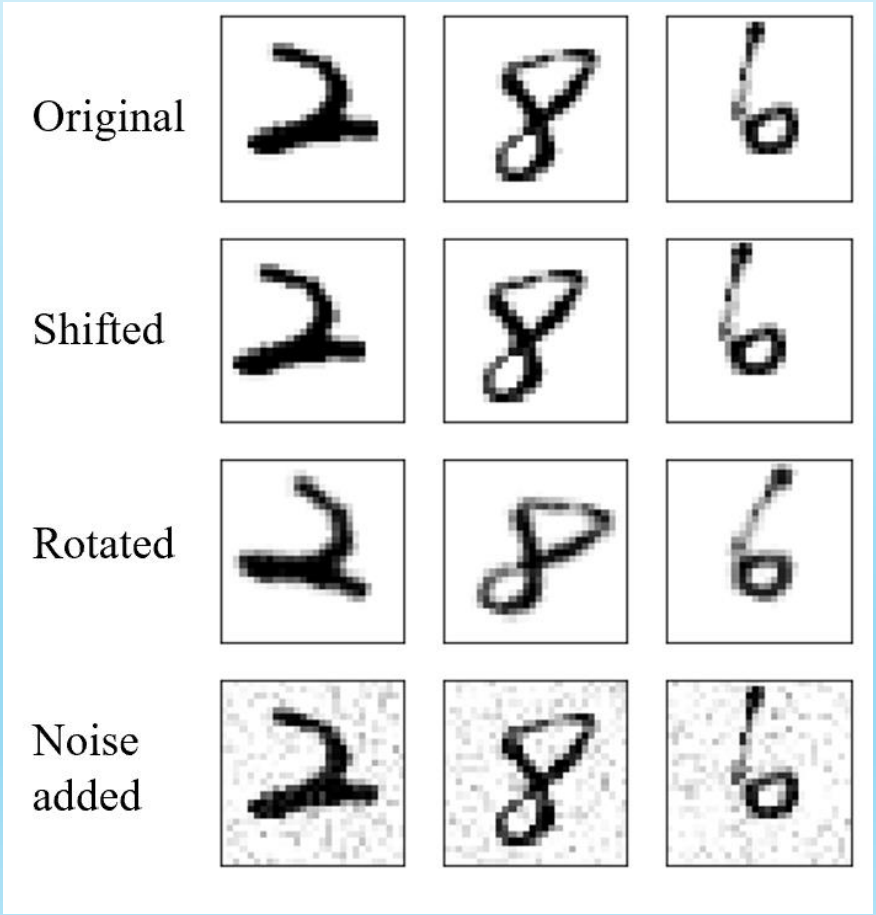
Visualization of error spaces for "0"

- Only one example remains uncovered by the predictions by three networks ($|E_{\mathrm{CNN}} \cap E_{\mathrm{RF}} \cap E_{\mathrm{SVC}}| = 1$)



Input data diversity

Using CNN with perturbated data for prediction to the labels of digits



Moves the digit to left by two pixels

Rotates the digit by twenty degrees in the clockwise direction

Uses Gaussian-distributed additive noise with 0.01 variance

Number of classification errors

The classification errors increase by data perturbation in most cases

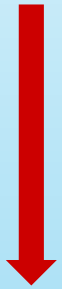
- Interestingly, however, there are some cases where the errors are reduced
  - i.e., for label 5 and 8 with added noise

Label	0	1	2	3	4	5	6	7	8	9	Total
$\lvert E_{CNN,o} \rvert$	3	6	11	3	5	9	22	11	11	28	109
$\lvert E_{CNN,s} \rvert$	35	85	58	18	20	21	52	18	32	54	393
$\lvert E_{CNN,r} \rvert$	5	47	70	19	105	24	104	147	57	113	691
$\lvert E_{CNN,n} \rvert$	8	8	11	3	6	8	29	17	9	29	128

Increased coverage of errors

The coverage of errors can increase just by using perturbated data

$\mathrm{Cov}(\mathrm{CNN}, \{o\})$	0.9891
$\mathrm{Cov}(\mathrm{CNN}, \{o, s\})$	0.9930
$\mathrm{Cov}(\mathrm{CNN}, \{o, s, r, n\})$	0.9957



increase

Classification of traffic sign images

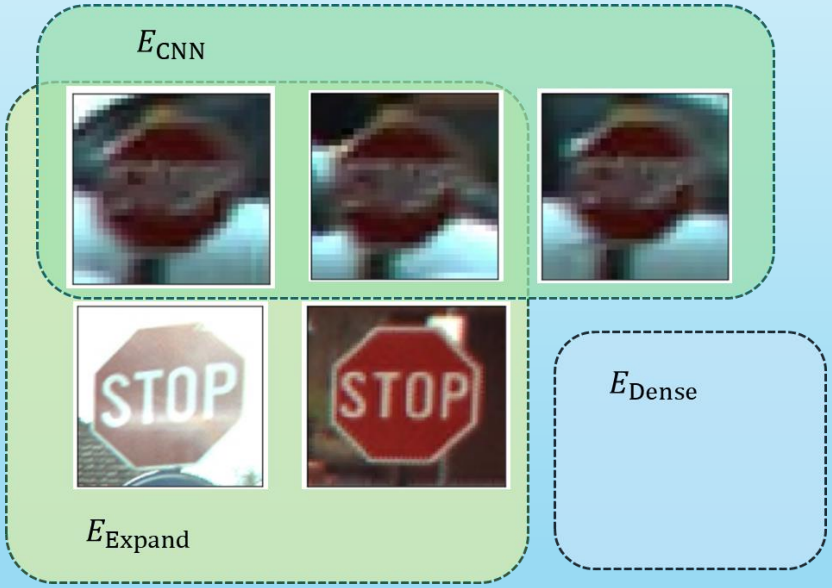
Not all label predictions are equally important

Classifications of "Stop", "No entry" and "No stop" are particularly important

Errors by three neural networks

The coverages of errors for "Stop", "No entry" and "No stop" reach 1.0

Label	Stop	No entry	No stop	Total
$\lvert S \rvert$	45	61	11	2520
$\lvert E_{\mathrm{CNN}} \rvert$	3	0	1	130
$\lvert E_{\mathrm{Dense}} \rvert$	0	0	0	247
$\lvert E_{\mathrm{Expand}} \rvert$	4	0	0	157
Cov(CNN)	0.9333	1.0000	0.9091	0.9484
Cov(CNN, Expand)	0.9556	1.0000	1.0000	0.9619
Cov(CNN, Dense, Expand)	1.0000	1.0000	1.0000	0.9746



Interestingly, for this specific task, Dense network contributes to increase the coverage of errors

Outline

1. Quality issue of Machine Learning (ML) systems
2. Diversity of ML models
3. Experimental study
4. **System reliability model and analysis**
5. Related work
6. Conclusion

System reliability model and analysis

To address RQ2, we propose the reliability model for 3-version ML architecture

- System reliability
  - The probability that the system output is correct in terms of input data from the real world application context
  - Is **NOT** equal to the accuracy on the test data set (which only gives an empirical estimate of the reliability)
- Objective
  - providing a reliability model to estimate the reliability of 3-version ML architecture using diversity metrics

Reliability model for 3-version system

Redundancy with independently fail modules and majority vote

- System reliability by majority voting from 3 outputs

$$R_{NV}(3) = R_1R_2 + R_1R_3 + R_2R_3 - 2R_1R_2R_3.$$

where R_i is the reliability of component *i*'s output

- When each component reliability is equivalent to *R*, it is the reliability of triple module redundancy (TMR) system

$$\mathrm{TMR} = 3R^2 - 2R^3$$

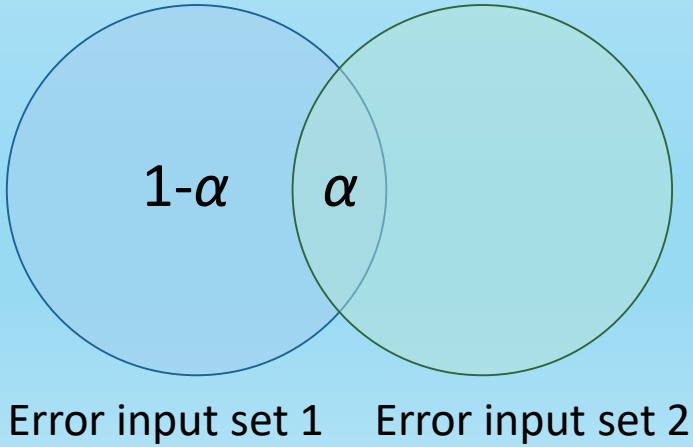
Reliability model for 3-version system

Redundancy with dependent fail modules and majority vote

- The reliability of an N-version programming system

$$R_{NV\alpha}(\alpha, 3) = 1 - \alpha(3 - 2\alpha)(1 - R)$$

where α is the similarity percentage of error input sets



Reliability model with diversity

Incorporating the diversity measure to the reliability model for 3-version system

- Intersection of error spaces

$$\alpha_I := \frac{\left|\bigcap_{i\in I} \mathcal{E}_i\right|}{|\mathcal{S}|}$$

Error space of m_i

Total sample space

- The reliability model for 3-version architecture using ML modules m_1, m_2, m_3

$$R_{3Vd}(m_1, m_2, m_3) = 1 - \left(\alpha_{\{1,2\}} + \alpha_{\{1,3\}} + \alpha_{\{2,3\}} - 2\alpha_{\{1,2,3\}}\right)$$

Empirical diversity and reliability estimation

Empirical estimates of the diversity measures are used for estimating reliability

- The system reliability of 3-version architecture

For MNIST

Module reliability	R_{CNN}	0.9891
	R_{RF}	0.9707
	R_{SVM}	0.9704
Empirical diversity	$\hat{\alpha}_{\{\mathrm{CNN,RF}\}}$	0.7523
	$\hat{\alpha}_{\{\mathrm{CNN,SVM}\}}$	0.6697
	$\hat{\alpha}_{\{\mathrm{RF,SVM}\}}$	0.5802
	$\hat{\alpha}_{\{\mathrm{CNN,RF,SVM}\}}$	0.6055
System reliability	$R_{3Vd}(\mathrm{CNN}, \mathrm{RF}, \mathrm{SVM})$	0.9807
	$R_{NV}(3)$	0.9985
	TMR	0.9984
	$R_{NV\alpha}(\hat{\alpha}_{\{\mathrm{CNN,RF}\}}, 3)$	0.9738

Overestimate

Underestimate

Condition for reliability improvement

- When the reliability of 3-version architecture competes the best ML module reliability?

$$R_{3Vd}(m_1, m_2, m_3) - R_1 > 0$$

- When $\left|\varepsilon_1 \cap \overline{\varepsilon_2} \cap \overline{\varepsilon_3}\right| - \left|\overline{\varepsilon_1} \cap \varepsilon_2 \cap \varepsilon_3\right| > 0$ holds, 3-version architecture achieves the higher reliability
  - By the test samples, we can empirically estimate the values of the terms in the condition

Related work

Multi-version ML approaches have been studied in different contexts and purposes

- Multi-version ML approaches in
  1. Generating a better machine learning model in terms of accuracy
  2. Testing an implementation of machine learning algorithm
  3. Improving the reliability of the system using machine learning models

Conclusion

- Our findings from the experiments and reliability analysis can be summarized in the following system design guide

Exploiting input diversity

- The approach using perturbated input data can be easily introduced for diversifying the outputs

Using multi-version models for error detection

- If any disagreement occurs among the multiple prediction results, we can discard the prediction results for safety

Evaluating the effectiveness of 3-version architecture

- Our necessary condition can give a guide to evaluate the effectiveness of 3-version architecture

Thank you!