大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

機械学習工学研究会
MACHINE LEARNING SYSTEMS ENGINEERING

QAML

ＡＩプロダクト
品質保証
コンソーシアム

Challenges in Quality Assurance for Machine Learning-based Systems

Fuyuki Ishikawa, National Institute of Informatics, Japan
with QAML Project / JSSST-MLSE / QA4AI Consortium

f-ishikawa@nii.ac.jp
http://research.nii.ac.jp/~f-ishikawa/en/

About Speaker

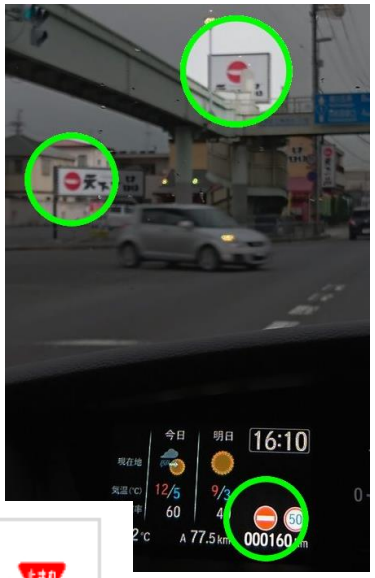
- Dependability: formal methods, testing, optimization, assurance arguments, etc.
- Current focus (1): dependability of (autonomous) driving systems *(talk at A-MOST yesterday)*
- *Current focus (2): dependability of machine-learning-based systems*
  - Chair of SIG-MLSE
  - Vice Steering Chair of QA4AI Consortium
  - A new research project just started

“Interesting” Examples

How Will You Test? (1)

- When Honda sees ramen shop sign
  - First buzz in Dec 2017

    [https://twitter.com/_gyochan_/status/938240168078622720]
    [https://twitter.com/Bleu_kakeru727/status/937680760491753473]



  - Now a caution on the web site

    http://www.honda.co.jp/hondasensing/feature/srf/

色や形の判別が
つきにくく、
対象の標識が無いのに、
標識を表示する

[http://www.tenkaippin.co.jp/company.html]

  - Second buzz in Sep 2018

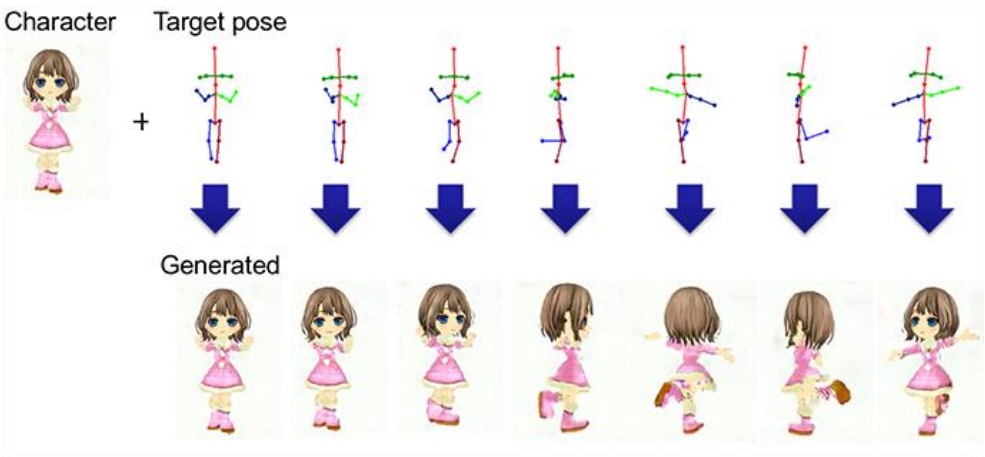
8:24 PM - 12 Sep 2018

15,099 Retweets 20,908 Likes

Can you find beforehand or prevent adverse (?) news??

How Will You Test? (2)

- From DeNA (May 2018)
  - Generate an image of a certain pose
  - Generate a movie given a pose sequence while changing the character



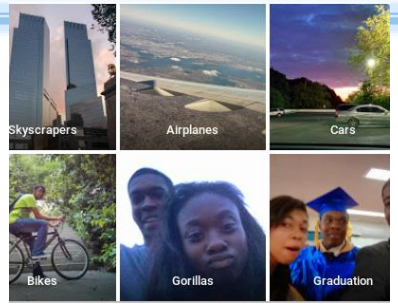
[https://dena.com/intl/anime-generation/]

What do you ensure to sell this to anime companies?

How do you Judge Acceptance?

- Technically unsolved issues
  - e.g., in Google Photo

    [https://www.theguardian.com/technology/2015/jul/01/google-sorry-racist-auto-tag-photo-app]
    [https://www.theguardian.com/technology/2018/jan/12/google-racism-ban-gorilla-black-people]



- Continuous learning, unclear oracle
  - A Microsoft bot learnt and made absolutely unacceptable tweets

    [https://www.nytimes.com/2016/03/25/technology/microsoft-created-a-twitter-bot-to-learn-from-users-it-quickly-became-a-racist-jerk.html]
  - A tweet by a Japanese bot rather attracted:

    "No way, you bald"

    "natural" from a "high-school girl"

    to her "parent" (Windows Japan) ?

    [https://twitter.com/ms_rinna/status/706615350524252160]

Correct Result vs. (Implicitly) Expected Logic

- A lot of studies on explainability (XAI)

➡ How do we use such techniques in engineering?

- Which part of the input image was used for the classification?

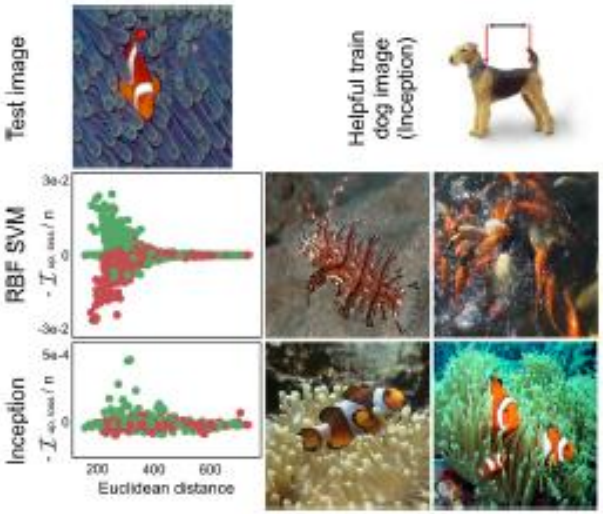
➡ *snow for wolf*

(a) Husky classified as wolf (b) Explanation

[Ribeiro et. al., " Why Should I Trust You?": Explaining the Predictions of Any Classifier, KDD'16]

- What samples in training had high influence on the classification?

➡ *fishes with similar colors*



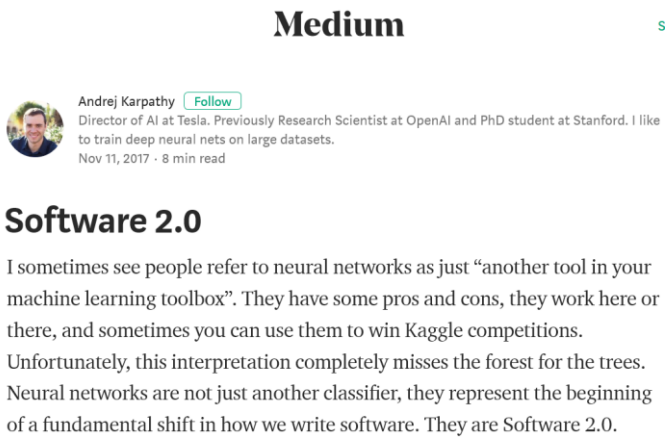
[Koh et. al., Understanding Black-box Predictions via Influence Functions, ICML'17]

Source of New Difficulties

Software 2.0 or Inductive Software Dev.

- Let us focus on ML
  - Present movement on AI was driven by ML, specifically advance in deep learning techniques
  - With ML, we construct a software component in a different way: derive the rule that governs the behavior from training data (not directly from engineers)
  - In the Japanese industry, the terms "inductive software development" and "inductive programming" is also used to clarify the essence

[https://medium.com/@karpathy/software-2-0-a64152b37c35]



Medium

Andrej Karpathy Follow
Director of AI at Tesla. Previously Research Scientist at OpenAI and PhD student at Stanford. I like to train deep neural nets on large datasets.
Nov 11, 2017 · 8 min read

Software 2.0

I sometimes see people refer to neural networks as just "another tool in your machine learning toolbox". They have some pros and cons, they work here or there, and sometimes you can use them to win Kaggle competitions. Unfortunately, this interpretation completely misses the forest for the trees. Neural networks are not just another classifier, they represent the beginning of a fundamental shift in how we write software. They are Software 2.0.

Example

Boundary created from training data

Gibbon

35 24 210	20 121 24	122 81 20
211 54 42	12 222 90	88 79 116
24 36 98	98 181 31	66 31 198

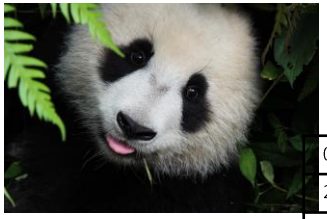
13 83 33	13 45 94	75 74 111
111 8 73	192 1 221	237 31 1
74 35 122	93 76 244	73 211 45

0 245 210	20 12 114	84 99 100
11 86 99	121 88 91	18 0 77
46 87 121	70 76 122	122 14 94

Panda

254 32 67	222 88 1	108 76 14
12 86 222	98 75 122	111 74 74
198 87 33	188 173 4	68 176 83

77 81 123	122 158 6	76 63 42
3 3 78	19 183 84	76 63 123
98 83 111	123 7 99	253 48 91

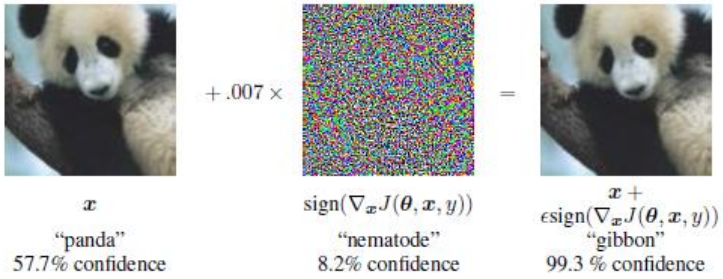


0 24 31	20 21 124	12 101 50
21 54 242	112 22 90	8 79 214
124 56 85	98 99 141	166 1 198

[http://free-photos.gatag.net/]

Impacts on Testing (Not Comprehensive)

- Large black-box to handle various situations
  - Principle of unit testing (and debugging) invalidated
- Errors as normal behavior (i.e., not 100% accuracy)
  - Test fail does not mean existence of a bug
- Often non-testable [Weyuker, On Testing Non-Testable Programs, 1982]
  - No oracle or oracle too costly to build
- Unknown boundaries in the implementation, adversarial examples
  - Little confidence on how it works in "similar" cases (no "equivalence classes")



[Goodfellow et al., Explaining and Harnessing Adversarial Examples, 2015]

V&V Research Emerging in SE Community

- SMT-based verification [CAV'17]
- Search for non-robust cases with "neuron coverage" [SOSP'17] [ICSE'18]
- System-level falsification [NFM'17]
- Safe reinforcement learning [AAAI'18]
- Verification by stochastic game [TACAS'18]
- Metamorphic testing case studies [ISSTA'18]
- Empirical study on bug statistics [ISSTA'18]
- Mutation analysis [ISSRE'18]
- Updated coverage criteria [ASE'18]
- Fairness testing [ASE'18]
- Concolic testing [ASE'18]
- Surprise-driven testing [ICSE'19]
- Demonstration: structural coverage is misleading [ICSE'19]
- (and more) e.g., [https://github.com/TrustAI/Literature-on-DNN-Verification-and-Testing]

Movement in Japan (Partial)

機械学習工学研究会
MACHINE LEARNING SYSTEMS ENGINEERING

- SIG-MLSE (Apr 2018)
  - SIG on Machine Learning Systems Engineering
  - Providing venues for presentations and discussions
  - Events almost every month, often go over the capacity
- QA4AI Consortium (Apr 2018)
  - Consortium of Quality Assurance for Artificial-Intelligence-based products and services
  - Guidelines for QA (to be released in May)
- And many others …
  - e.g., AIST + NII :
    Standards for quality like ISO26262 or Common Criteria

ＡＩプロダクト
品質保証
コンソーシアム

Guideline from QA4AI (to appear)

- 5 axis of evaluation
  - Model, System, and Data
  - Process: especially agility to improve and react to unexpected outcomes
  - To match with Customer Expectation
- Technical catalog (Body Of Knowledge)
- Domain-specific insights
  - Autonomous driving, factory data processing, smart speakers, and creative generation

Can only be the current best and to be updated

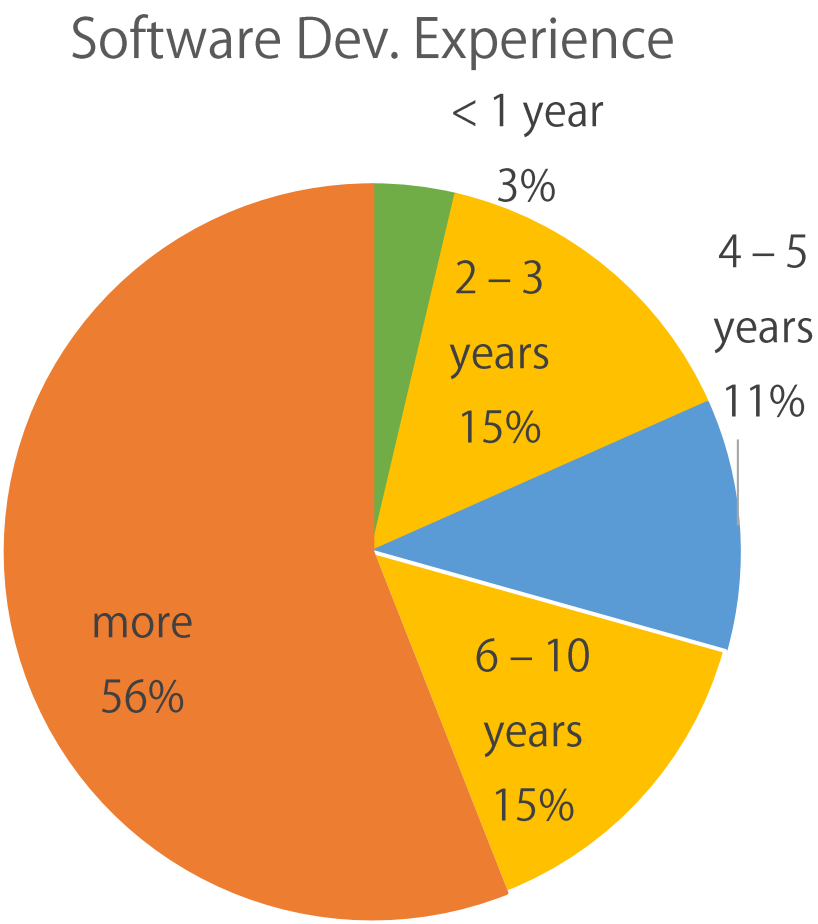
Perception by Engineers

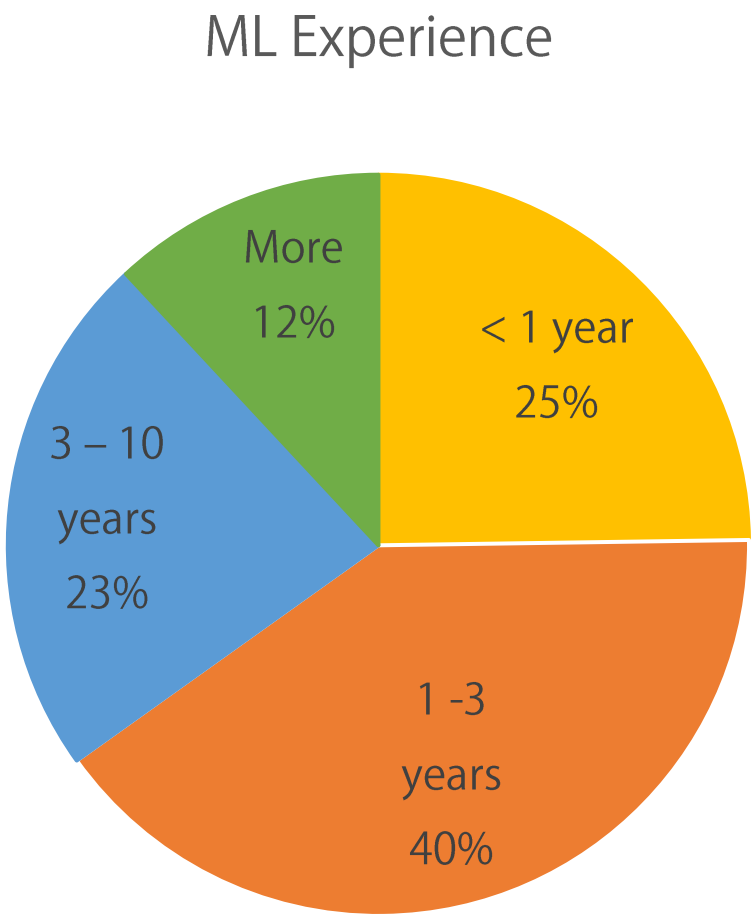
[Ishikawa et al., How do Engineers Perceive Difficulties in Engineering of Machine-Learning Systems? – Questionnaire Survey, CESSER-IP'19]

Questionnaire Survey

- Method
  - Dissemination by mailing lists and social networks (software engineering, ML, and AI)
  - “those who have used ML at work”
  - 280 answers
- Question aspects
  - Experience on SE activities and on ML techniques
  - Past projects that used ML
  - Quality attributes considered significant
  - Perception of difficulties
  - Characteristics of ML that lead to the difficulties

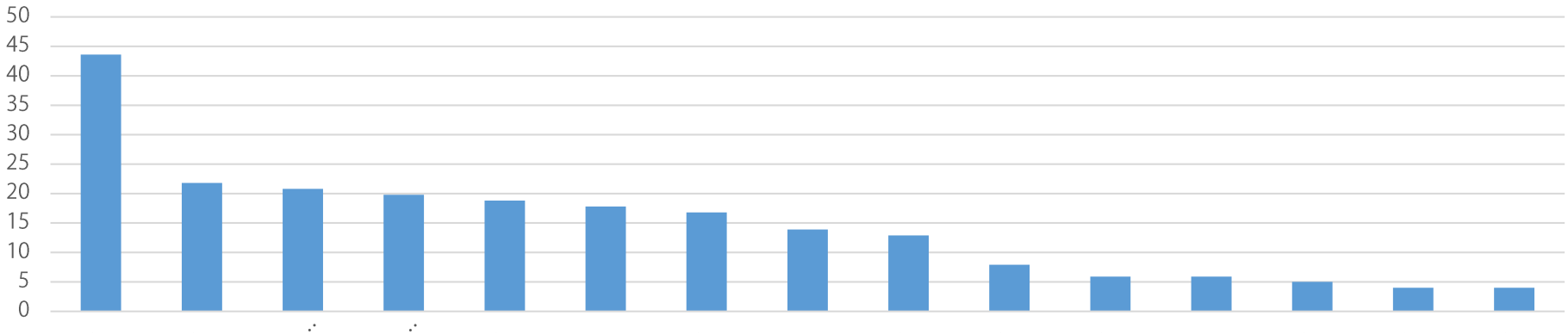
Profiling





Experienced engineers were recently pushed to learn and use ML

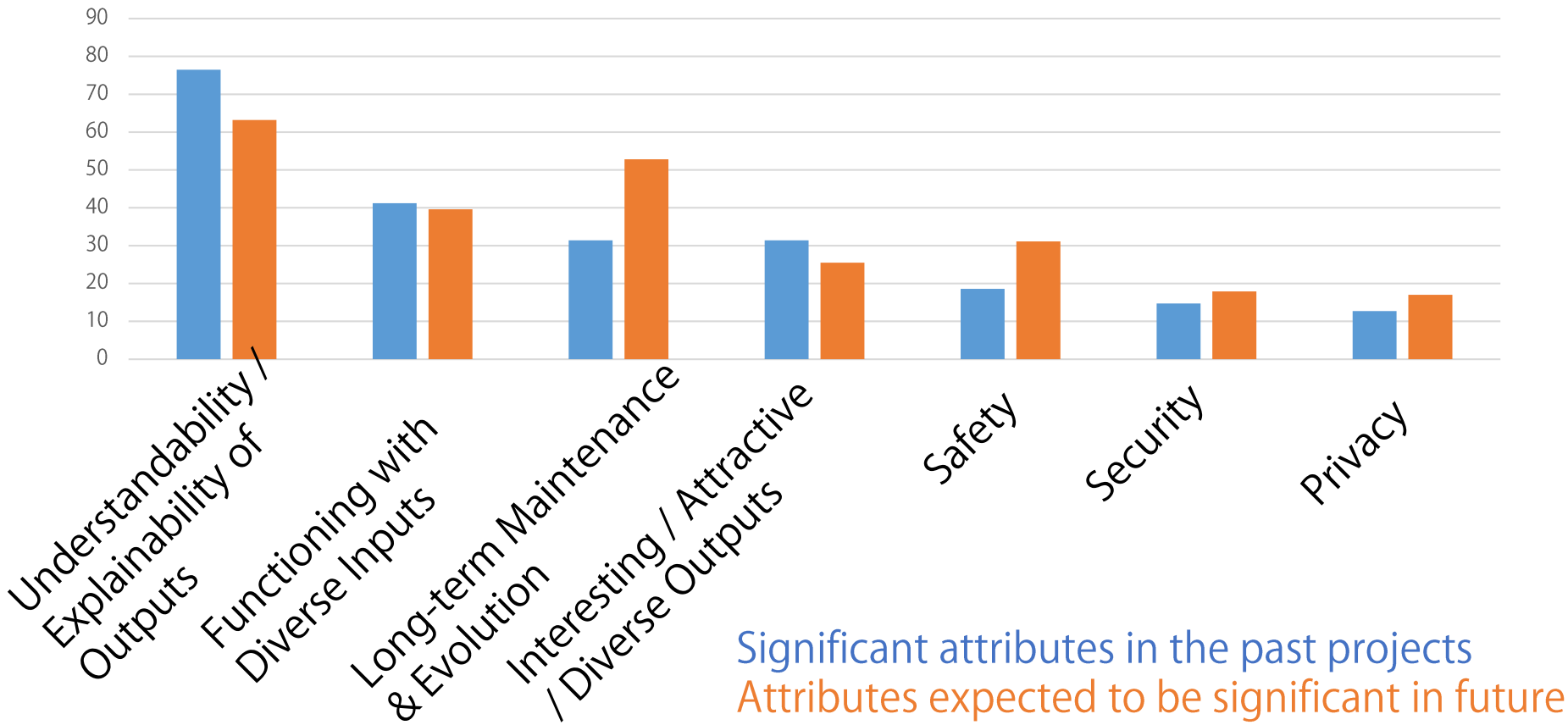
ML Usage Domain



Manufacturing
Information & Communication
Company-Specific Services
Foundational Development (e.g., middleware)
Academic Research
Automotive / Railway
Finance / Insurance
Home Appliance
Wholesale and Retail
Education
Life Service / Amusement
Medical / Welfare
Construction
Real Estate
Utility

Application in manufacturing is large (in Japan)

Significant Quality Attributes



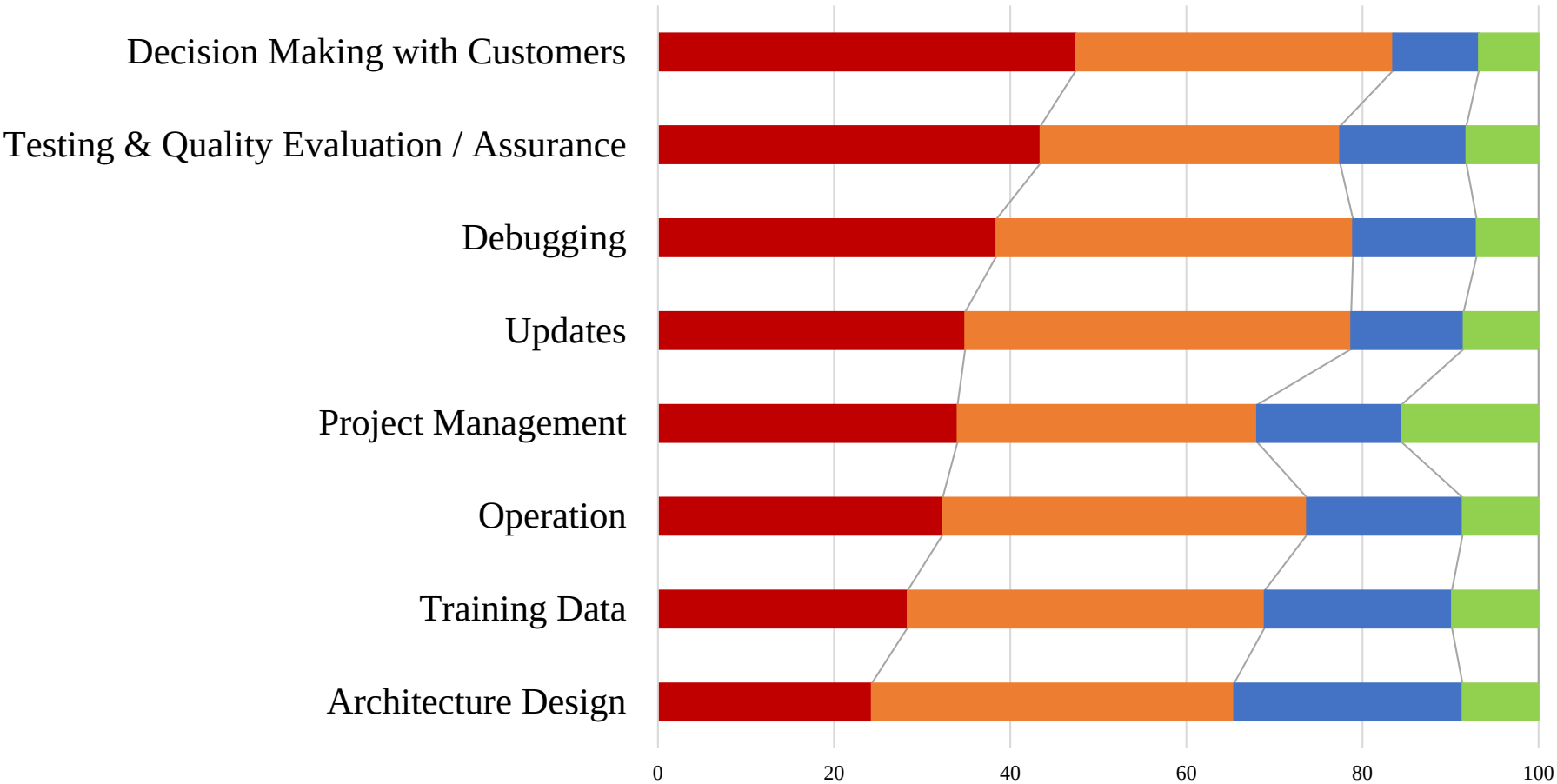
XAI (eXplainable AI) is thought as significant also in practice

Maintenance, security, and privacy are somewhat left behind, or domains were limited not to consider them

Difficulty Level

We need to use new approaches as the existing ones do not work anymore

We can apply the same approaches but methods, etc., are still immature



We already have dedicated methods, etc.

We can use existing methods, frameworks, or tools

Some Typical Comments

- Tough negotiation with customers
  - We need to make a priori agreement that we cannot make a priori agreement on what we can develop, e.g., what accuracy we will be able to achieve
  - Free attacks at the acceptance time (with too much expectation on “AI”)
- “POC (Proof-of-Concept) Poverty”
  - There are so many projects that finish at the POC phase
- “We have data! We have! fraud”
  - Having large volume of data does not mean it is well annotated, not noisy, or with good correlations

Challenges more than Testing

Quality Attributes

- From SQuaRE Metrics (ISO/IEC 25023:2016)
  - e.g., Functional coverage: what proportion of the specified functions has been implemented?

    1 - #functions_missing / #functions_specified

  ➡ *Whether/how should we interpret and define similar metrics for ML-based systems?*

  - e.g., Functional correctness: what proportion of functions provides the correct results?

    1 - #functions_incorrect / #functions_considered

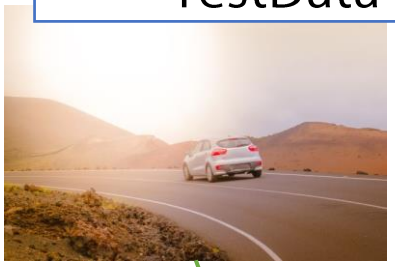
  ➡ *Is the accuracy (and the like) as the average for the whole data space enough?*

Our Simple Example: Attributed Tests

- To specify & check constraints
- To describe current status
- To discuss validity
- To compare with operational data

We tested with 100,000 data!

What data … ? Did you test misty days?



carCameraImg1 : TestData

atr1 : Attribute

atr2 : Attribute

weatherReport : AttributionMethod

middle : Confidence

situation : AttributeType

sunrise : AttributeValue

weatherCondition : AttributeType

cloudy : AttributeValue

imgSet1 : TestDataSet

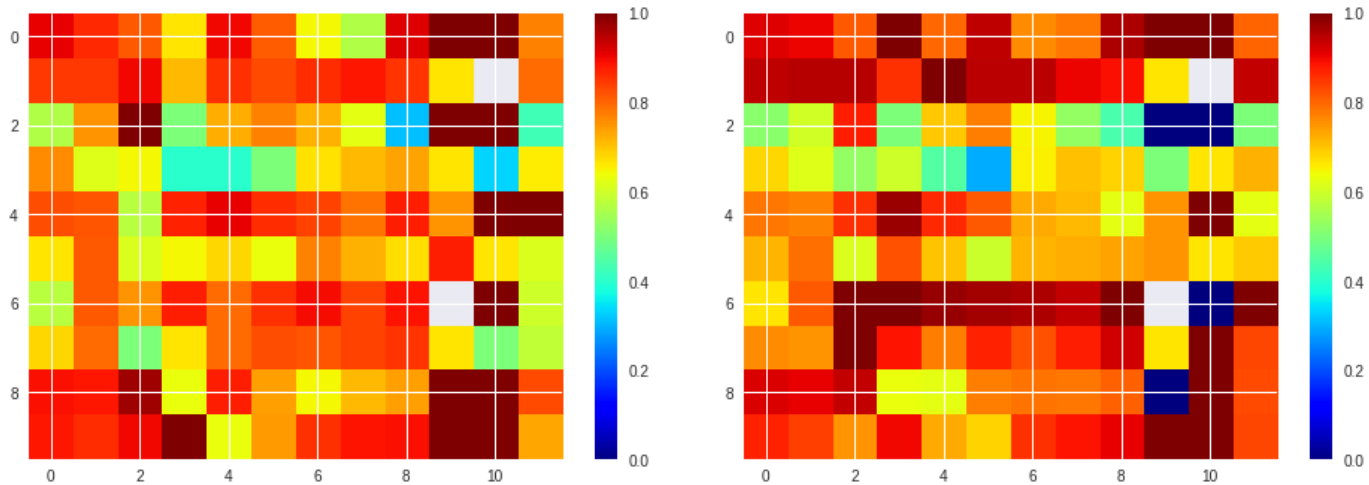
#misty = 2000 : DataSetProperty

Actual representation may be spreadsheet, GUI, or whatever

[Ishikawa, Concepts in Quality Assessment for Machine Learning - From Test Data to Arguments, ER'18]

Our Simple Example: Attributed Tests

- What if we do "equivalence classes" testing based on requirements and environmental assumptions?



Accuracy for different classification targets (e.g., "car") in the vertical axis and different primary colors (e.g., "red") in the horizontal axis

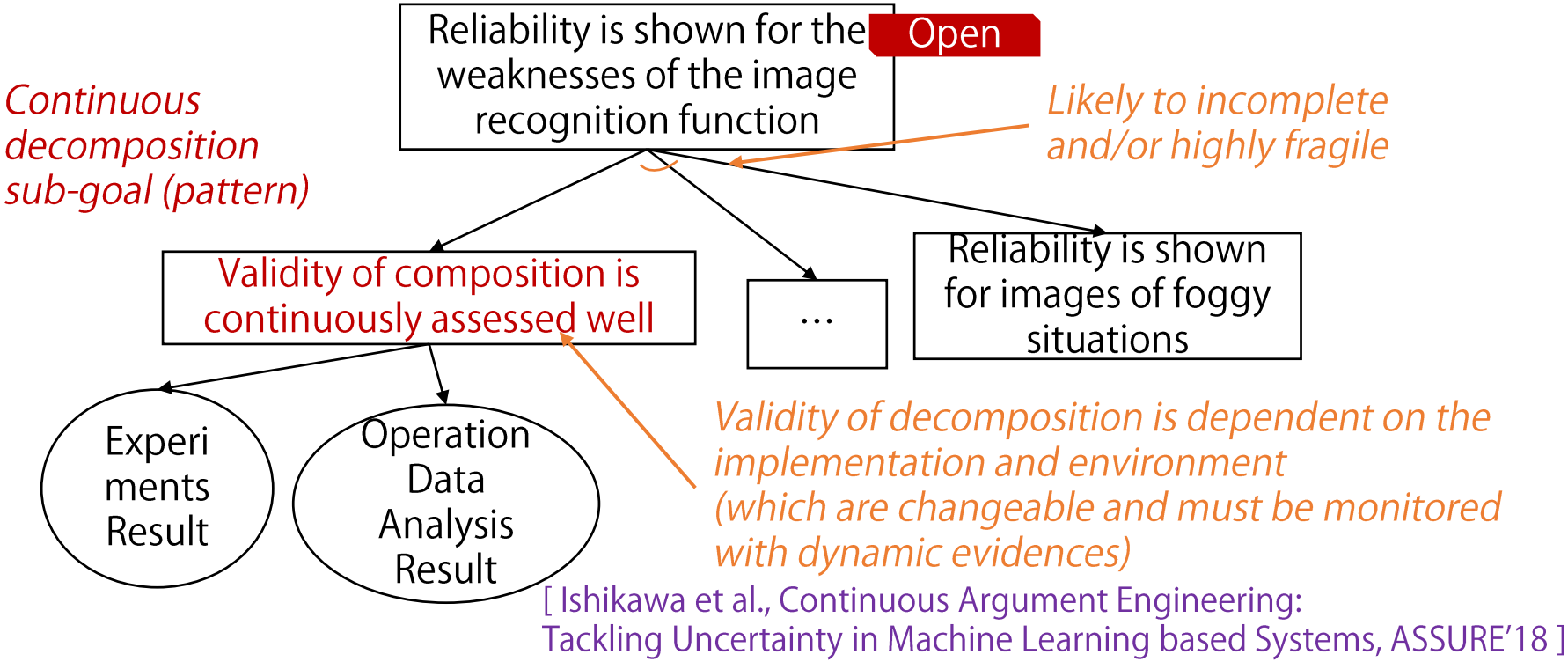
- Left : before data augmentation (accuracy: 79%)
- Right : after data augmentation (accuracy: 81%)

(keras sample of Capsule CNN for CIFAR10 dataset)

Weakness changes!

Our Example: Continuous Arguments Eng.

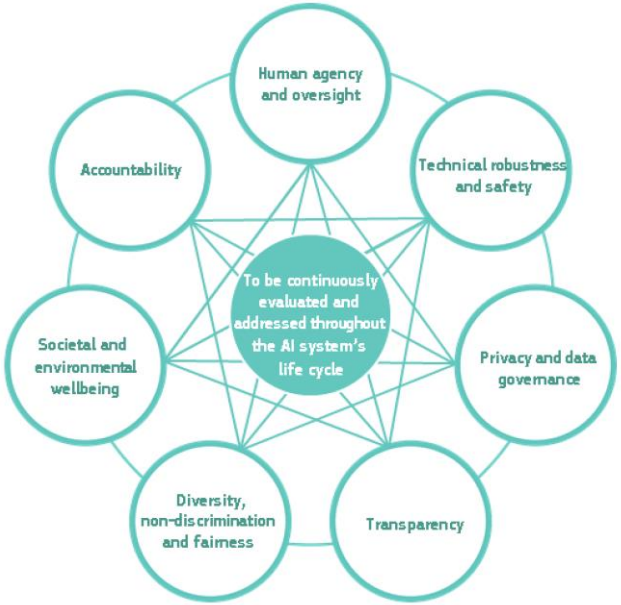
- Explicitly model intrinsic uncertainty in arguments to be aware of risk and need for continuous update
  - Three patterns for uncertainty in goal decomposition, evidence contribution, and feasibility of goals



[Ishikawa et al., Continuous Argument Engineering: Tackling Uncertainty in Machine Learning based Systems, ASSURE'18]

Encoding into Engineering Terminology

- How do we “encode” the required principles?
- Assessment list in EU AI Ethics
  - e.g., could the AI system affect human autonomy by interfering with the (end) user’s decision-making process in an unintended way?
  - e.g., did you take safeguards to prevent overconfidence in or overreliance on the AI system for work processes?



[https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai]

FAQs on Emerging Testing Research (1)

- Metamorphic testing

  Define the expected output via a metamorphic relation: *"changing an input this way will change the output that way"*

  - e.g., switching the RGB channels and re-learning will not change the classification results

[Dwarakanath, Identifying Implementation Bugs in Machine Learning, 2018]

How to find such relations?

This is not testing based requirements: how much should we do? How to argue cost-effectiveness?

FAQs on Emerging Testing Research (2)

When to use what? What is the whole picture and roles of each method?

- Standard measurement of accuracy
- Search for adversarial examples
- Metamorphic testing
- Neuron coverage
- …

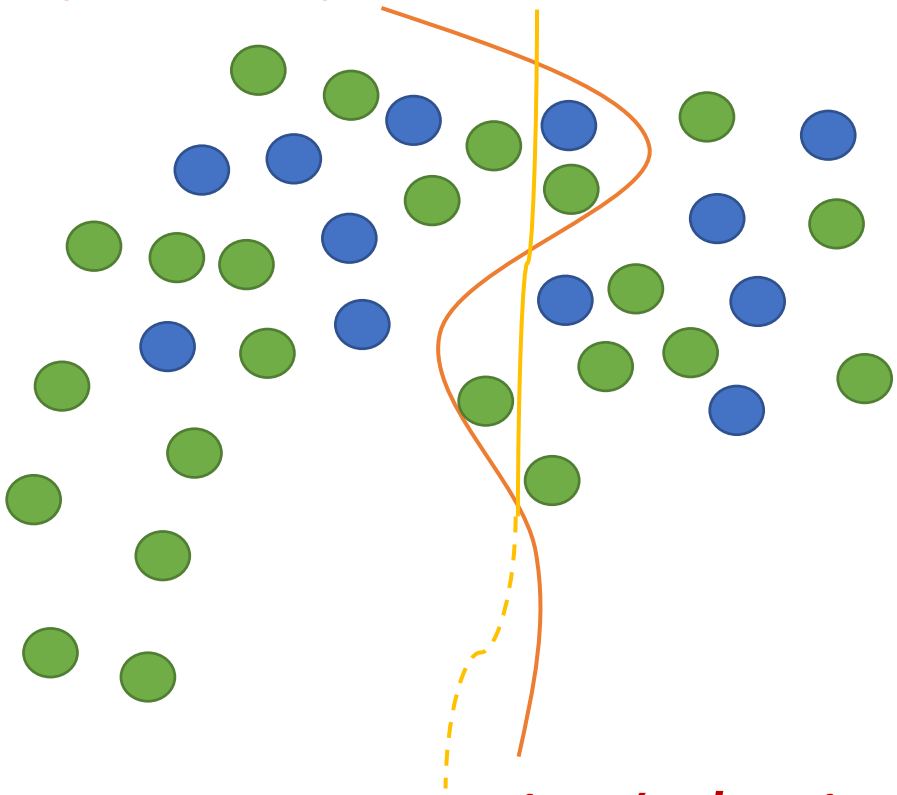
And, is this all of what are required/necessary/effective?

Rough Illustration

In the input space

Implemented boundary (unknown)

Required/correct boundary (often unclear, vague, and implicit)



Current test dataset

Data to appear in operation i.e., what really matters (unknown beforehand)

Where are you generating/selecting new test data?
What coverage/division are you considering?

Example: White-box Structural Coverage

"How diverse output values of each neuron have been observed by a test suite?"

1.1 original 1.2 with added rain

- Originally used for increasing diversity in the "bad" (non-robust) outputs to be generated

  [Pei et al., DeepXplore: Automated Whitebox Testing of Deep Learning Systems, 2017]

  [Tian et al., DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars, 2018]

- Used as criteria to judge a test suite (dataset)

  [Ma et al., DeepGauge: multi-granularity testing criteria for deep learning systems, 2018]

- Now said "could be misleading"
  - e.g., dependency rather on the search methods
  - e.g., natural faults vs. adversarial faults

  [Li et al., Structural Coverage Criteria for Neural Networks Could Be Misleading, 2019]

Summary

Machine learning as a new development paradigm

- Different characteristics, invalidating some of existing principles (in testing and in software engineering)
- Currently, efforts are initial and explorative:
  based on (implicit) hypothesis that we should do similar things to what we did for classical software systems

More contributions by the testing community demanded for essential discussions and insights on what are "good tests", what are "bugs", …