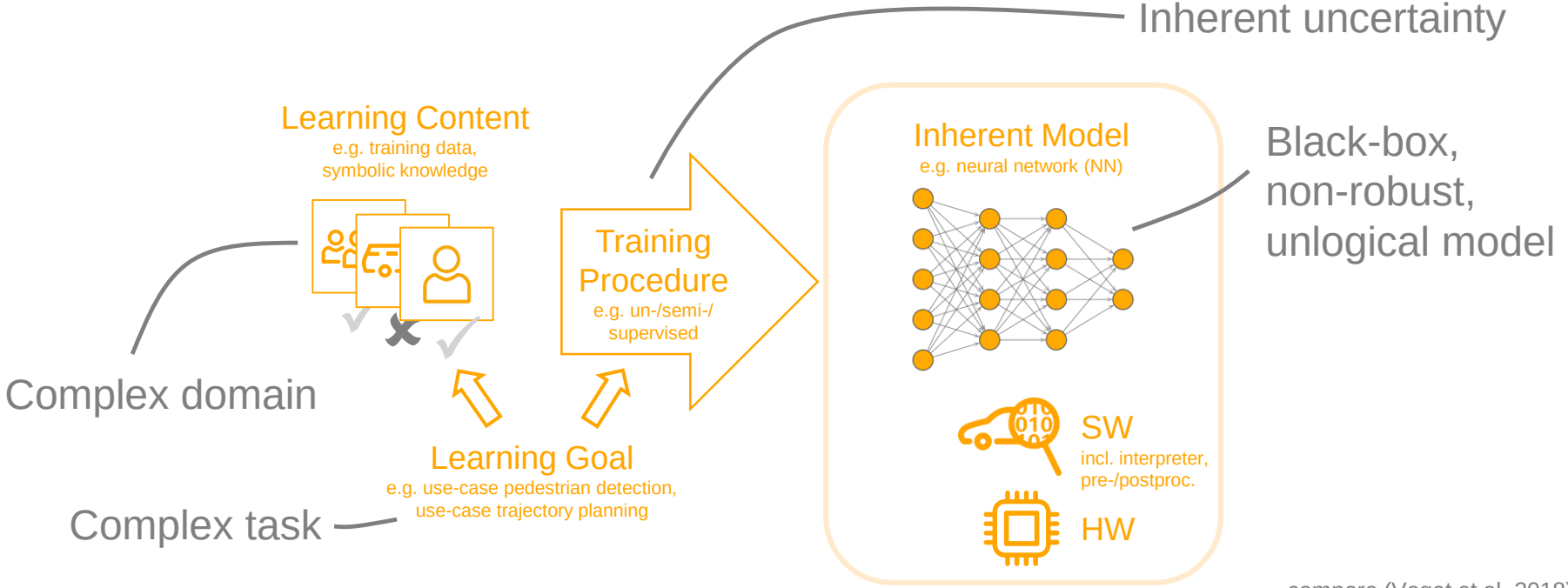
Continental
The Future in Motion

Safety Assurance of Machine Learning Based Systems

A Method Survey

www.continental-corporation.com

What is a Machine Learning (ML) Based System?
And what is new?



compare (Voget et al. 2018)

Life-Cycle Phases

1. **Requirements Engineering**
2. **Development**
3. **Model Verification and Validation**

G. Schwalbe, M. Schels © Continental AG

Requirements Engineering
Overall Goal

absence of unreasonable risk
3.132 in ISO 26262 Standard

Perform *safely*
within the specified **domain**
when integrated into the **system**
wrt. available **experience**.

Requirements Engineering
Performance Requirements

Perform *safely*
within the specified **domain**
when integrated into the **system**
wrt. available **experience**.

- **Black-box**: specialized performance measures,
  e.g. detection performance for occluded objects (Cheng et al. 2018)
- **White-box**: specifics of machine learning (ML) method,
  e.g.
  - **Robustness**,
    e.g. adversarial robustness (Katz et al. 2017)
  - **Plausibility** of environment model and logic,
    e.g. respecting laws of physics

Requirements Engineering
Knowledge Specification

Perform *safely*
within the specified **domain**
when integrated into the **system**
wrt. available **experience**.

For both training and testing:

- **Data representativity**:
  - **scenario** coverage (Cheng et al. 2018),
    e.g. applied to input space ontology (Bagschik et al. 2018)
  - **experience** coverage
  - **model behavior** coverage

Requirements Engineering
System Requirements

Perform *safely*
within the specified **domain**
when integrated into the **system**
wrt. available **experience**.

Increase system fault tolerance

- **Runtime monitoring** = plausibility/validity checks
  - Task specific, *e.g. based on domain specific rules* (Shalev-Shwartz et al. 2017) *or maps*
  - Model specific, *e.g. uncertainty monitoring*
- **Model redundancy**, see ISO 26262, (E) 7.4.12
  ⚠ Requires model diversity measure!

Requirements Engineering
Experience based Requirements

Perform *safely*
within the specified **domain**
when integrated into the **system**
wrt. available **experience**.

Requirements from experience in

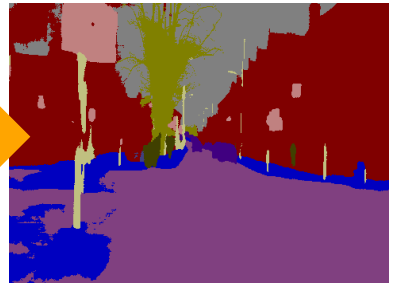
- **Domain**, *e.g. known physics, corner cases*
- **Model**, *e.g. known limitations, previous failures*

Life-Cycle Phases

1. **Requirements Engineering**
2. **Development**
   - **2.1** Uncertainty Treatment
   - **2.2** Knowledge Insertion
   - **2.3** Robustness Enhancement
3. **Model Verification and Validation**

G. Schwalbe, M. Schels © Continental AG

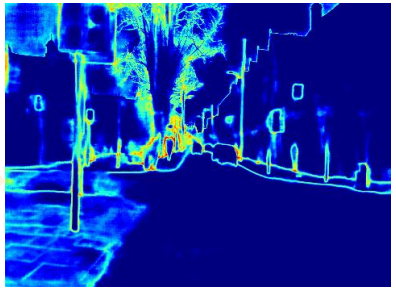
Development
Uncertainty Treatment



ML is statistical!

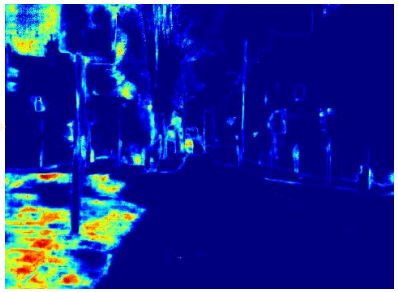
- **Extract** the uncertainty
- **Use** the uncertainty
  - Runtime monitoring
  - Propagation through system

ML uncertainties:



Aleatoric

Uncertainty in data
(sensor noise, motion)



Epistemic

Uncertainty about model

(Kendall and Gal 2017), Fig. 1, p. 2

Development
Inclusion of Expert Knowledge

- **Data**: describe via examples,
  e.g. adversarial attacks, safety critical corner cases
- **Optimization objective or topology** (Wang 2018): Include
  - Intermediate steps / needed concepts
  - Rules
  - Safe states

Concept “head” embedded in neural network
(Bau et al. 2017), Fig. 9

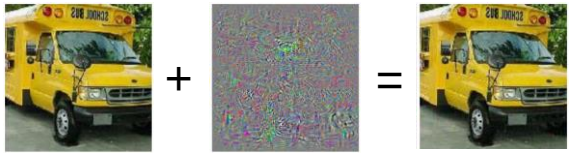
Development
Robustness Enhancements

result indifferent to slight input changes: $|x - y|_{in} < e_{in} \Rightarrow |f(x) - f(y)|_{out} < e_{out}$

⚠ Many ML models not robust = locally chaotic!

- **Data**:
  - **Augmentation** by adversaries
  - **Removal** of adversarial features
- **Training**: regularization, uncertainty treatment, …

Examples of NN adversaries:



“school bus” “ostrich”

(Guo et al. 2018), Fig. 1, p. 2

“speed limit 45”

(Eykholt et al. 2018), Tab. 1

Life-Cycle Phases

1 **Requirements Engineering**

2 **Development**

3 **Model Verification and Validation**

- **3.1** How to access model internals?
- **3.2** How to prove model internals?

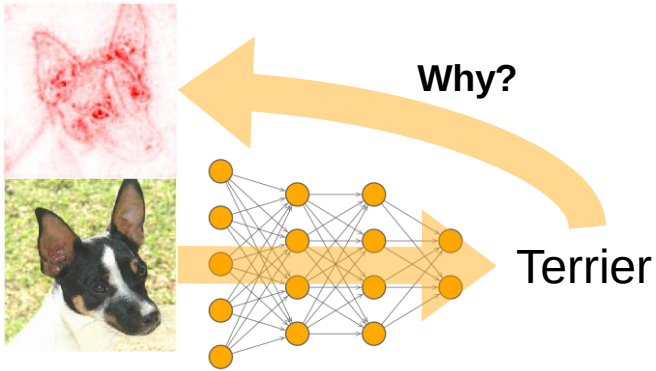
Verification and Validation
Qualitative Analysis Methods

Additional explanatory output
e.g. hierarchical information
(Roychowdhury, Diligenti, and Gori 2018)

Feature visualization

Activation pattern of one neuron
(Olah, Mordvintsev, and Schubert 2017)

Attention analysis



(Kindermans et al. 2018), Fig. 6

Verification and Validation
Quantitative Analysis Methods

- **Sub-task** & **representation** analysis

- **Rule extraction**

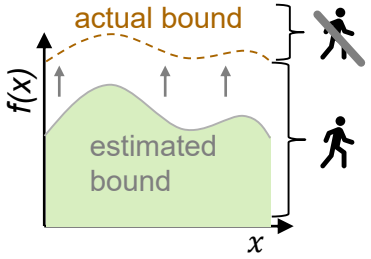
Scarce!

Verification and Validation
Proving Methods

- **Testing** (test data representativity!)
- **Formal verification**
  e.g. for NNs:
  - Solvers
  - Output bound estimation
  - Search algorithms

Equation system:

$$P_{child} > \epsilon \Rightarrow P_{human} > \epsilon$$



Outlook
Current Challenges

ML specialties:

- Data driven
- Inherently uncertain
- Black-box logic
- Non-robust

Biggest challenges:

- **Data representativity** measures
- Methods for
  - **Expert knowledge inclusion**
  - **Quantitative model analysis**

Thanks for listening!

Contact: Gesina.Schwalbe@continental-corporation.com

References I

Bagschik, G., T. Menzel, and M. Maurer. 2018. “Ontology Based Scene Creation for the Development of Automated Vehicles.” In *Proc. 2018 IEEE Intelligent Vehicles Symp.*, 1813–20. Changshu, Suzhou, China: IEEE. https://doi.org/10.1109/IVS.2018.8500632.

Bau, David, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. “Network Dissection: Quantifying Interpretability of Deep Visual Representations.” In *Proc. 2017 IEEE Conf. Comput. Vision and Pattern Recognition*, 3319–3327. Honolulu, HI, USA: IEEE Computer Society. https://doi.org/10.1109/CVPR.2017.354.

Cheng, Chih-Hong, Georg Nührenberg, Chung-Hao Huang, Harald Ruess, and Hirotoshi Yasuoka. 2018. “Towards Dependability Metrics for Neural Networks.” In *16th ACM/IEEE Int. Conf. Formal Methods and Models for System Design*, 43–46. Beijing, China: IEEE. https://doi.org/10.1109/MEMCOD.2018.8556962.

Eykholt, Kevin, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. “Robust Physical-World Attacks on Deep Learning Visual Classification.” In *Proc. 2018 IEEE Conf. Computer Vision and Pattern Recognition*, 1625–1634. Salt Lake City, UT, USA: IEEE Computer Society. https://doi.org/10.1109/CVPR.2018.00175.

Fong, Ruth, and Andrea Vedaldi. 2018. “Net2Vec: Quantifying and Explaining How Concepts Are Encoded by Filters in Deep Neural Networks.” In *Proc. 2018 IEEE Conf. Comput. Vision and Pattern Recognition*, 8730–8738. Salt Lake City, UT, USA: IEEE Computer Society. https://doi.org/10.1109/CVPR.2018.00910.

Guo, Jianmin, Yu Jiang, Yue Zhao, Quan Chen, and Jiaguang Sun. 2018. “DLFuzz: Differential Fuzzing Testing of Deep Learning Systems.” In *Proc. ACM Joint Meeting on European Software Engineering Conf. and Symp. Foundations of Software Engineering*, 739–743. Lake Buena Vista, FL, USA: ACM. https://doi.org/10.1145/3236024.3264835.

References II

Katz, Guy, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. 2017. “Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks.” In *Proc. 29th Int. Conf. Comput. Aided Verification*, 97–117. Lecture Notes in Computer Science. Springer International Publishing. http://arxiv.org/abs/1702.01135.

Kendall, Alex, and Yarin Gal. 2017. “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?” In *Advances in Neural Information Processing Systems 30*, 5580–5590. Long Beach, CA, USA. http://papers.nips.cc/paper/7141-what-uncertainties-do-we-need-in-bayesian-deep-learning-for-computer-vision.

Kim, Been, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. 2018. “Interpretability beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV).” In *Proc. 35th Int. Conf. Machine Learning*, 80:2668–77. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm, Sweden: PMLR. http://proceedings.mlr.press/v80/kim18d.html.

Kindermans, Pieter-Jan, Kristof T. Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. 2018. “Learning How to Explain Neural Networks: PatternNet and PatternAttribution.” In *Proc. 6th Int. Conf. on Learning Representations*. Vancouver, Canada. https://openreview.net/forum?id=Hkn7CBaTW.

Olah, Chris, Alexander Mordvintsev, and Ludwig Schubert. 2017. “Feature Visualization.” *Distill* 2 (11): e7. https://doi.org/10.23915/distill.00007.

Roychowdhury, Soumali, Michelangelo Diligenti, and Marco Gori. 2018. “Image Classification Using Deep Learning and Prior Knowledge.” In *Workshops of the 32nd AAAI Conf. Artificial Intelligence*, WS-18:336–343. AAAI Workshops. New Orleans, Louisiana, USA: AAAI Press. https://aaai.org/ocs/index.php/WS/AAAIW18/paper/view/16575.

Shalev-Shwartz, Shai, Shaked Shammah, and Amnon Shashua. 2017. “On a Formal Model of Safe and Scalable Self-Driving Cars.” *CoRR* abs/1708.06374. http://arxiv.org/abs/1708.06374.

References III

Voget, Stefan, Alexander Rudolph, and Jürgen Mottok. 2018. “A Consistent Safety Case Argumentation for Artificial Intelligence in Safety Related Automotive Systems.” In *Proc. 9th European Congress on Embedded Real Time Systems*. Toulouse, France. https://www.erts2018.org/uploads/program/ERTS_2018_paper_13.pdf.

Wang, Hu. 2018. “ReNN: Rule-Embedded Neural Networks.” In *Proc. 24th Int. Conf. Pattern Recognition*, 824–829. Beijing, China: IEEE Computer Society. https://doi.org/10.1109/ICPR.2018.8545379.