|  |  |  |  |
| --- | --- | --- | --- |
| ITU Logo | INTERNATIONAL TELECOMMUNICATION UNION  **TELECOMMUNICATION STANDARDIZATION SECTOR**  STUDY PERIOD 2017-2020 | | FG-AI4H-E-025 |
| **ITU-T Focus Group on AI for Health** |
| **Original: English** |
| **WG(s):** | | N/A | Geneva, 30 May – 1 June 2019 |
| **DOCUMENT** | | | |
| **Source:** | | Fraunhofer HHI | |
| **Title:** | | Robustness - Safety and reliability in AI4H | |
| **Purpose:** | | Discussion | |
| **Contact:** | | Wojciech Samek Fraunhofer HHI Germany | Tel: +49 30 31002-417 Fax: +49 30 31002-190 Email: [wojciech.samek@hhi.fraunhofer.de](mailto:wojciech.samek@hhi.fraunhofer.de) |
| **Contact:** | | Vignesh Srinivasan Fraunhofer HHI Germany | Tel: +49 30 31002-187 Fax: +49 30 31002-190 Email: [vignesh.srinivasan@hhi.fraunhofer.de](mailto:vignesh.srinivasan@hhi.fraunhofer.de) |
|  | | Luis Oala Fraunhofer HHI Germany | Tel: +49 30 31002-104 Fax: +49 30 31002-190 Email: [luis.oala@hhi.fraunhofer.de](mailto:luis.oala@hhi.fraunhofer.de) |
|  | | Thomas Wiegand Fraunhofer HHI Germany | Tel: +49 30 31002-617 Fax: +49 30 31002-190 Email: [thomas.wiegand@hhi.fraunhofer.de](mailto:thomas.wiegand@hhi.fraunhofer.de) |

|  |  |
| --- | --- |
| **Abstract:** | Safety and reliability are important concerns when contemplating the use of AI systems in medical applications. In machine learning similar concerns are discussed under the term robustness, which has its origin in classical statistical theory and finds new interpretations in the light of novel technologies like deep learning. We provide an introduction to the concept and explain the two potential sources of robustness risk. In addition, we unify tools from different machine learning discourses and organize them into four action areas for robustness enhancement. We show how these tools can be integrated into the life cycle of AI systems to make them safer and more reliable. Finally, we give recommendations as to what criteria an AI system should fulfil to attain enhanced robustness. |

# Introduction

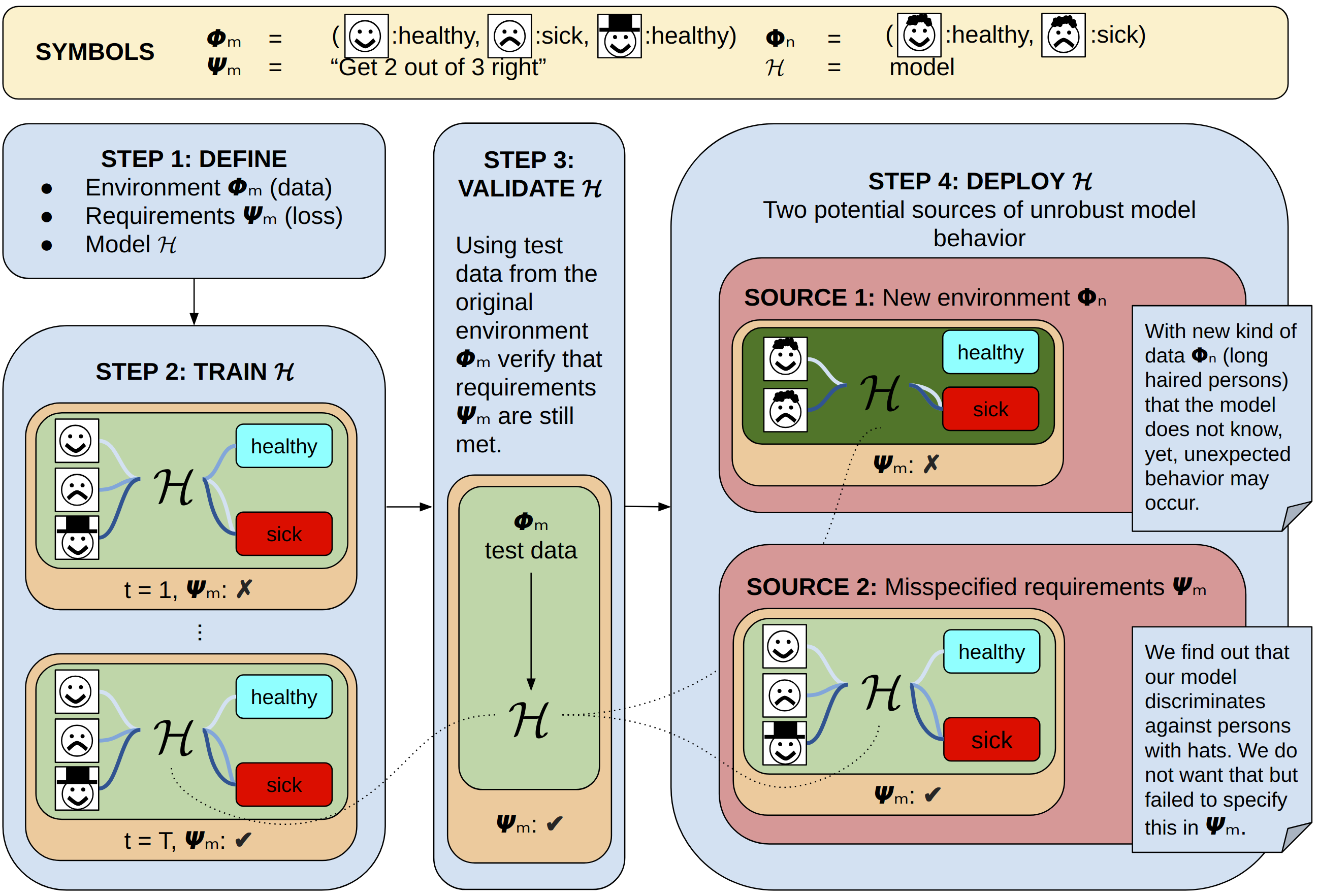
Modern AI systems based on deep learning, reinforcement learning or hybrids thereof are powerful technologies. They are also fickle technologies whose behaviour is often hard to fathom. This creates a risk for system failure which is of particular concern when attempting to deploy AI systems in health applications.

Decades of machine learning research has produced a number of tools to contain risks on robustness. Many of today’s research efforts around deep and reinforcement learning attempt to find improved ways of doing so. In this report we provide a high-level illustration of the two potential sources of robustness risks for AI systems. In addition, we identify four action areas along the life cycle of AI systems for mitigating robustness risks. While we are not claiming completeness we note that a breadth of tools are covered, the time-tested alongside the very recent. We hope this report provides meaningful concepts and categories for facilitating an informed and interdisciplinary discussion of AI system robustness in the context of health applications.

# The AI System Life Cycle and Robustness

The life cycle of an AI system can be organized into for general steps which are visualized in *Illustration 1*. The first step comprises defining an AI system. This includes the choice of a model

H, a training environment *Φ*(data) and requirements *Ψ* (e.g. optimization objective, evaluation metrics). Then in the second step the model H is trained until it fulfils the specified requirements *Ψ*. After the training has concluded the model is typically validated in step three. After successful validation the model can then be considered for deployment in step four.

  
Illustration 1: The life cycle of an AI system and the two possible sources of failing robustness. In Step 1 the environment Φm (training data), requirements Ψm (optimization objective, evaluation metrics) and model H for the AI system are decided upon. Then in step 2 the AI system is trained until the requirements Ψm are fulfilled. Step 3 comprises validating the AI system on test data from the same distribution as Φm. Finally, in step 4 the model is deployed. During deployment two possible sources for robustness failure can be identified. Source 1 constitutes changes in the environment, i.e. a new type of data the model has not seen during training. Source 2 comprises misspecifications of the requirements that were used for model training, e.g. we might have failed to account for the fact that no discrimination should take place on account of a person wearing a hat or not.

In the words of Peter J. Huber robustness can broadly be understood as “insensitivity to small deviations from the assumptions” [1]. While useful this definition appears too narrow for the setting we often find ourselves in with modern AI systems. It is often not even known what assumptions can be made when working deep learning AI systems. Thus, a broader definition is needed. Thomas G. Dietterich advances a robustness view that distinguishes between how an AI system deals with known unknowns and how it behaves with respect unknown unknowns [2]. For this report we utilize the following working definition of robustness that merges these views and emerges from the ideas of [3]: robustness is a desideratum we place on an AI system to not commit any gross, unexpected errors under slight changes of the environment *Φ* or to at least handle them benignly, e.g. by letting a human AI system operator know that something unusual has happened.

As we later explain in detail contributions to enhanced model robustness can be made at each step of the AI system life cycle. Robustness during step four, deployment, is of particular concern when considering the application of AI systems. Following the analysis of [3] robustness failures at the deployment step can originate from two potential sources:

* First, it is possible that the environment *Φn*in which a model operates is different from the one *Φm* it was calibrated in. For example, it has been shown that standard convolutional neural networks used for image classification are not invariant to common perturbations like blurring [4].
* Second, it may turn out that the requirements *Ψm* we originally specified were insufficient to capture some behaviour we actually care about. This might mean we need to come up with new evaluation metrics to capture the erroneous behaviour and make it visible. For example, new metrics are actively being researched to avoid racial and gender bias in image classification models [5].

# Four Action Areas for Enhancing Robustness

We grouped the available tools to enhance the robustness of an AI system into four action areas along the life cycle steps. As visualized in Illustration 2 these four groups comprise data fidelity, robust training, robustness validation and alarm systems. In the following we will explain these in detail and list available tools for each action area (see Table 1 for an overview of all tools).

**Data fidelity** can be understood as imposing desiderata on the data that is used for training an AI system. This can take the form of diversity criteria to mandate a balance with respect to certain features like age, socioeconomic status or race. This can take the form of datasheets as proposed by [6]. Datasheets would summarize a datasets key statistics along with usage recommendations and aspects that users of this dataset should be aware of. Another, and often used, data fidelity tool is preprocessing and normalization. This is commonly used to ensure that input data during deployment lie in the same range as during training or to satisfy certain modelling assumptions, e.g. uncorrelated inputs. Popular tools include zero centering data (each input dimension will have a mean of zero), principal component analysis (commonly used for decorrelating data) [7], whitening (scaling decorrelated data to unit variance) [7], standardization (to normalize scales across dimensions) and min-max scaling (normalize data to ranges [-1,1] or [0,1]).

**Robust training** comprises a group of methods that help exposing an AI system to changes in the data environment that are likely to otherwise induce robustness failures during deployment. In this way the AI system can be seen as “getting used” to the types of environment changes that would otherwise cause it to break. An important tool in this action area is adversarial training [8,9,10,11] which aims at reducing an AI system’s vulnerability to adversarial examples: data points that to humans are visually indistinguishable from original inputs but the AI system nevertheless misclassifies. Under adversarial training such examples are included in the training data so that the AI system can learn to treat them correctly. Currently, the most popular method follows [12]. Another strategy to achieve this is by employing generative models. Generative algorithms [13,14,15,16,17] model how the data was generated before classifying it. They are also an effective alternative for protecting AI systems against adversarial attacks [18,19,20,21,22]. However, most of these methods have been found not to work effectively at protecting the AI system classifier. An attacker can specifically target the weakness of the reconstruction algorithm and craft an adversarial example for the AI system classifier [23]. This problem can be alleviated by employing Langevin Dynamics (LD) [24]. Finally, robust training can be improved by employing stability training. The aim of stability training [25] is to improve robustness against data distortions without compromising classification performance. Instead of using distortion instances in the training data, stability training generates images that are disturbed by Gaussian noise and feeds the images to the network at the same time as the reference samples. The network then has the following task: make the outputs for the disturbed image similar to the outputs of the reference images. This implicitly forces a restriction on the sensitivity of a model to small perturbations in the input data.

**Robustness validation** features a set of tools aimed at verifying the performance of an AI system. This can include new data environments, evaluating the model under new requirement metrics or putting it through a stress test for specific edge cases and vulnerabilites. A simple strategy for obtaining a less noisy estimate of an AI system ’s predictive performance is so called cross-validation. Under this evaluation regime the available data Φ is partitioned into groups. The AI system is then trained on each of the groups, alternating the test group for each training run for all possible choices of a test group [26]. A major drawback of this approach is the computational burden it incurs for models with expensive training procedures which oftentimes is the case for deep learning AI systems. In the realm of statistical scholarship hypothesis testing forms an important methodological pillar. Many of the popular data modelling approaches in this field have been studied for decades and, owing to their analytic accessibility, in many cases their behaviors are well understood. The ordinary least squares (OLS) estimator is such a well studied approach which boasts a plethora of tests to better interpret the resulting model. This includes tests for hypotheses on individual regression coefficients, e.g. the so called t-test, or linear combinations of hypotheses, e.g. the F-test, [27] as well as tests for properties like conditional heteroskedasticity [28] or serial autocorrelation [29,30]. This level of model understanding and interpretation has not carried over to deep learning based AI systems. An important reason for this absence is that deep learning AI systems typically do not lend themselves to the type of analytical treatment that is possible with hallmark approaches from classical statistics. Some theoretically motivated model selection criteria that have been carried over from statistical theory to the deep learning setting include variations on the log evidence like Akaikes’s information criterion (AIC) [31], Schwarz’s Bayesian information criterion (BIC) [32] or the Occam-weighted likelihood used in Bayesian model selection [33]. Furthermore, adversarial vulnerability tests can be used to simulate attacks on the AI system. There are several attacking strategies developed to pose a threat to a AI system. Almost all of them follow the principle that the classification should be changed with only minimal modification of the input. Projected Gradient Descent (PGD) [12] is in its core the fundamental version of a first-order attack. Other attacking strategies like Carlini-Wagner (CW) [34], Momentum Iterative Method (MIM) [35], Elastic-Net Attack against DNN (EAD) [36] or Fast Gradient Sign Method (FGSM) [37] can be considered to be a variation of this attack. Finally, new requirements can be introduced to probe the trained AI system. These new metrics can for example be drawn from insights of so called fair, accountable, transparent (FAT) AI research which offers various approaches to formalizing fairness and biases [38,39,40,41,42,43]. There also exist proposals how to incorporate such FAT measurements in AI system training and applications [44,45,46,47,48] which can be utilized in the robust training action area. Lastly one should note that FAT research has already produced a number of software repositories aimed at benchmarking an algorithm’s susceptibility to problems of bias and fairness, for example AI Fairness 360 [49], Python Fairness Package [50], or TuringBox [51].

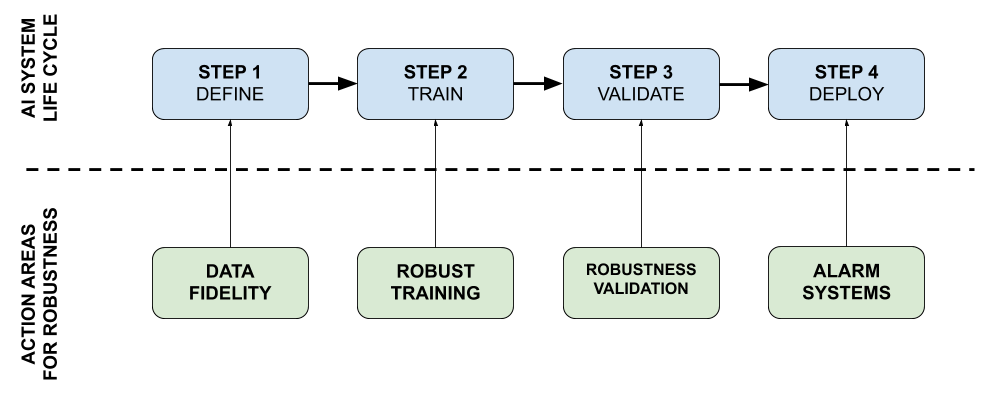
  
Illustration 2: The four steps of an AI system life cycle alongside four possible action areas for enhanced robustness. We can impose requirements on the fidelity of the data, e.g. by restricting the type of data a model can take as input. Another strategy is to design the training procedure in a way that robustness enhancing methods are being used, e.g. adversarial training. In addition, we can use the validation step in the AI system work flow to probe the model in new environments or under new requirements. Finally, we can mandate the use of different alarm systems that indicate to a AI system operator when the AI system is confronted with an unknown situation during deployment.

Table 1: Summary overview of robustness enhancing tools per action area.

| Action areas | Tools |
| --- | --- |
| Data fidelity | * Data diversity [6] * Preprocessing (zero centering, PCA [7], Whitening [7]) * Normalization (standardization, min-max scaling) |
| Robust training | * Adversarial training [12] * Generative methods [18,19,20,21,22, 24] * Stability training [25] * FAT optimization objectives [44,45,46,47,48] |
| Robustness validation | * Cross-validation [26] * Classical tests (e.g. t-test, F-test [27], serial autocorrelation [29, 30]) * Information criteria (AIC [31], BIC [32], Occam-weighted likelihood [33]) * Vulnerability tests (PGD [12]) * FAT validation metrics [38,39,40,41,42,43] * FAT validation toolboxes [49, 50, 51] |
| Alarm systems | * Outlier tests (generative methods [52,53], Bayesian uncertainty [62,66]) * Attribution methods (gradient\*input [54], integrated gradients [55], layer-wise relevance propagation [56]/deep Taylor decomposition [57], perturbation-based attribution [58]) * Uncertainty quantification (Gaussian Processes [62], aleatoric uncertainty for deep learning AI systems [67], epistemic uncertainty for deep learning AI systems [66]) |

**Alarm systems** are important to monitor the AI system during deployment. Their purpose is to alert an AI system operator when something unusual is happening. Outlier tests are a case in point. Such tests signal when input deviates strongly from the types of input the model has been trained on. Generative models can be used to detect outliers for a given data distribution [52,53]. Any input lying on the manifold of the training data distribution will be given a high score by the generative AI system, as the model has seen data from this manifold during training. Conversely, an input which is very different from the training data will be given a low score, meaning that it is an outlier. Attribution methods can also be used. Formally, attribution methods typically deal with methods that aim to map input features to relevance scores that reflect the features’ contribution to the output of a model. As an alarm system these methods can be used to signal when the model bases its decision on input features very different from the ones a medical expert would use. Popular schemes include gradient\*input [54], integrated gradients [55], layer-wise relevance propagation [56]/deep Taylor decomposition [57] or perturbation-based attribution [58]. Finally, uncertainty quantification methods may be employed to signal uncertainty in the inputs, so called aleatoric uncertainty, uncertainty in the model decision, so called epistemic uncertainty, or unfamiliar inputs, which is related to epistemic uncertainty. Classic Bayesian modelling with Gaussian Processes provides built in epistemic uncertainty estimates [59,60,61,62,63]. As deep learning AI systems are not as amenable to an analytic treatment as Gaussian Processes numerous approximating treatments have surfaced [64, 65, 66]. A popular epistemic uncertainty quantification scheme for deep learning AI systems, called Monte Carlo dropout, was proposed by [66]. Aleatoric uncertainty quantification for deep learning AI systems has already been sketched out as early as 1994 by [67].

# Recommendations

To enhance the robust performance of AI systems in a real-world application checks and safety measures, as presented above, should be incorporated at each step of the AI system life cycle. Below we provide a summarized check list for important tools in each step of the AI system work flow.

Ensuring **data fidelity** can help to enhance the robustness of AI systems. Through preprocessing and normalization the input data can be brought into a shape that accommodates modelling assumptions such as decorrelated inputs or inputs in a certain value range. Thus, an AI system should include the following data fidelity protocols

* If possible restrict input data during deployment to be similar to input data during training
* Ensure that input data adheres to the modelling assumptions by e.g. zero centering, PCA, whitening, standardization and min-max scaling
* Evaluate and ensure the diversity of the data as per the requirements of the specific task, e.g. racial diversity

The robustness of AI systems can be enhanced by employing **robust training** protocols such as adversarial training, generative models or stability training. Thus, it should be ensured the following robust training tools were used

* Adversarial training using PGD attacks to shield against adversarial vulnerability
* Generative models
* Stability training to shield against common data perturbations

AI systems obtained after robust training should also undergo rigorous **robustness validation** before going into deployment. Thus, the following tools should be in place for the validation step

* Cross-validation
* If the model allows: hypothesis testing
* Adversarial and perturbation stress tests
* Pending suitably labeled data FAT metrics should be employed to evaluate task specific requirements that go beyond the original optimization objectives

**Alarm systems** are critical to making sure that failures are sufficiently signaled by the AI system. Any error or malfunction should be caught with the help of techniques available. To this end the following tools should be in place

* Ensure that an outlier test for input data is in place
* Include integrated gradients, or another attribution methods of your choice, so that a human can verify the basis of the AI system decisions
* Include an uncertainty quantification tool, like Monte Carlo dropout or an aleatoric proxy variable, so that the AI system ’s decision confidences are signalled to the AI system operator

# References

1. P. J. Huber. Robust statistics. *Wiley Series in Probability and Mathematical Statistics, New York: Wiley,* 1981.
2. T. G. Dietterich. Steps toward robust artificial intelligence. *AI Magazine*, *38*(3), 3-24, 2017.
3. Stuart Russell, Daniel Dewey, and Max Tegmark. Research Priorities for Robust and Beneficial Artificial Intelligence. *AI Magazine*, 36(4):105, December 2015.
4. Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261,* 2019.
5. Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pp. 77-91. 2018.
6. Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumeé III, and Kate Crawford. Datasheets for datasets.*arXiv preprint arXiv:1803.09010*, 2018.
7. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, New York, NY, USA, 2000.
8. Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.
9. Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.
10. Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. Adv-bnn: Improved adversarial defense through robust bayesian neural network. *arXiv preprint arXiv:1810.01279*, 2018.
11. Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. *arXiv preprint arXiv:1812.03411*, 2018.
12. Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.
13. Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.
14. Diederik P. Kingma, Tim Salimans, and Max Welling. Variational Dropout and the Local Reparameterization Trick. *CoRR*, abs/1506.02557, 2015.
15. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
16. Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016.
17. Rithesh Kumar, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energybased models. *arXiv preprint arXiv:1901.08508*, 2019.
18. Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, volume 9, 2018.
19. Andrew Ilyas, Ajil Jalal, Eirini Asteri, Constantinos Daskalakis, and Alexandros G Dimakis. The robust manifold defense: Adversarial training using generative models. *arXiv preprint arXiv:1712.09196*, 2017.
20. Shiwei Shen, Guoqing Jin, Ke Gao, and Yongdong Zhang. Ape-gan: Adversarial perturbation elimination with gan. *ICLR Submission, available on OpenReview*, 4, 2017.
21. Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.
22. Lukas Schott, Jonas Rauber, Wieland Brendel, and Matthias Bethge. Robust perception through analysis by synthesis. *arXiv preprint arXiv:1805.09190*, 2018.
23. Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.
24. Vignesh Srinivasan, Arturo Marban, Klaus-Robert Müller, Wojciech Samek, and Shinichi Nakajima. Counterstrike: Defending deep learning architectures against adversarial samples by langevin dynamics with supervised denoising autoencoder. *arXiv preprint arXiv:1805.12017*, 2018.
25. Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the Robustness of Deep Neural Networks via Stability Training. *arXiv:1604.04326 [cs]*, April 2016. arXiv: 1604.04326.
26. Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, NY, corrected at 8th printing 2009 edition, 2009. OCLC: 845772798.
27. Fumio Hayashi. *Econometrics*. Princeton Univ. Press, Princeton, 2000. OCLC: 247253903.
28. Halbert White. A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity. *Econometrica*, 48(4):817–838, 1980.
29. L. G. Godfrey. Testing Against General Autoregressive and Moving Average Error Models when the Regressors Include Lagged Dependent Variables. *Econometrica*, 46(6):1293–1301, 1978.
30. T. S. Breusch. Testing for Autocorrelation in Dynamic Linear Models\*. *Australian Economic Papers*, 17(31):334– 355, 1978.
31. H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716– 723, December 1974.
32. Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978.
33. David J. C. MacKay. *Bayesian methods for adaptive models*. phd, California Institute of Technology, 1992.
34. Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017.
35. Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. *arXiv preprint*, 2018.
36. Yash Sharma and Pin-Yu Chen. Breaking the madry defense model with *l*\_1-based adversarial examples. *arXiv preprint arXiv:1710.10733*, 2017.
37. Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
38. C. Dwork, M. Hardt, T. Pitassi, O. Reingold, & R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference* (pp. 214-226). ACM, 2012.
39. N. Papernot, P. McDaniel, A. Sinha, & M. Wellman. Towards the science of security and privacy in machine learning. Preprint at:[arXiv:1611.03814](https://arxiv.org/abs/1611.03814), 2016.
40. P. Gajane, & M. Pechenizkiy. On formalizing fairness in prediction with machine learning. Preprint at:[arXiv:1710.03184](https://arxiv.org/abs/1710.03184), 2017.
41. A. Weller. Challenges for Transparency. CoRR. Preprint at:[arXiv:1708.01870v1](https://arxiv.org/abs/1708.01870v1), 2017.
42. G. Yona & G. Rothblum. Probably Approximately Metric-Fair Learning. In *International Conference on Machine Learning* (pp. 5666-5674), 2018.
43. E. D. Moss. Translation Tutorial: Toward a Theory of Race for Fairness in Machine Learning. *In Proceedings of FAT\* conference (FAT\* Conference). ACM, New York, NY, USA, 2 pages,* 2019.
44. S. Shaikh, H. Vishwakarma, S. Mehta, K. R. Varshney, K. N. Ramamurthy, & D. Wei. An End-To-End Machine Learning Pipeline That Ensures Fairness Policies. *CoRR, abs/1710.06876*. Preprint at:[arXiv:1710.06876v1](https://arxiv.org/abs/1710.06876v1), 2017.
45. A. Paul, C. Jolley & A. Anthony. Reflecting the Past, Shaping the Future: Making AI Work for International Development. Retrieved from USAID:<https://www.usaid.gov/sites/default/files/documents/15396/AI-ML-in-Development.pdf>, 2018.
46. R. Dobbe & M. Ames. Translation Tutorial: Values, Reflection and Engagement in Automated Decision-Making. *In Proceedings of  ACM Conference on Fairness, Accountability, and Transparency (ACM FAT\* 2019). ACM, New York, NY, USA, Article 4, 2 pages,* 2019.
47. A. Albarghouthi, & S. Vinitsky. Fairness-Aware Programming. In *FAT\* ’19: Conference on Fairness, Accountability, and Transparency*, January 29–31, 2019, Atlanta, GA, USA. ACM, New York, NY, USA, 9 pages, 2019.
48. K. Holstein, J. W. Vaughan, H. Daumé III, M. Dudík, & H. Wallach. Improving fairness in machine learning systems: What do industry practitioners need?. *ACM CHI Conference on Human Factors in Computing Systems (CHI 2019).* Preprint at:[arXiv:1812.05239v2](https://arxiv.org/abs/1812.05239v2), 2018.
49. R. K. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, ... & S. Nagar. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. Preprint at:[arXiv:1810.01943v1](https://arxiv.org/abs/1810.01943v1), 2018.
50. S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, & D. Roth. A comparative study of fairness-enhancing interventions in machine learning. Preprint at:[arXiv:1802.04422v1](https://arxiv.org/abs/1802.04422v1), 2018.
51. Z. Epstein, B. H. Payne, J. H. Shen, C. J. Hong, B. Felbo, A. Dubey, ... & I. Rahwan. TuringBox: An Experimental Platform for the Evaluation of AI Systems. In *Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80*, 2018. (pp. 5826-5828), 2018.
52. Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017.
53. Nicholas Frosst, Sara Sabour, and Geoffrey Hinton. Darccc: Detecting adversaries by reconstruction from class conditional capsules. *arXiv preprint arXiv:1811.06969*, 2018.
54. Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. In *ICML*, 2017.
55. Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *ICML*, 2017.
56. Sebastian Bach, Alexander Binder, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Analyzing Classifiers: Fisher Vectors and Deep Neural Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2912–2920, 2016.
57. Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.
58. Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *ECCV*, 2014.
59. John S. Denker, Daniel B. Schwartz, Ben S. Wittner, Sara A. Solla, Richard E. Howard, Lawrence D. Jackel, and John J. Hopfield. Large Automatic Learning, Rule Extraction, and Generalization. *Complex Systems*, 1, 1987.
60. Geoffrey E. Hinton and Radford M. Neal. Bayesian Learning for Neural Networks. 1995.
61. Christopher K. I. Williams. Computing with Infinite Networks. In *NIPS*, 1996.
62. Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 3. print edition, 2008. OCLC: 552376743.
63. David Barber and Christopher M Bishop. Ensemble Learning in Bayesian Neural Networks. page 20.
64. Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.
65. Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. *arXiv:1505.05424 [cs, stat]*, May 2015. arXiv: 1505.05424.
66. Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *ICML*, 2016.
67. D. A. Nix and A. S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN’94)*, volume 1, pages 55–60 vol.1, June 1994.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_