Carnegie Mellon University

Software Engineering Institute

The Beginnings of AI Engineering

Thinking through how to build AI better

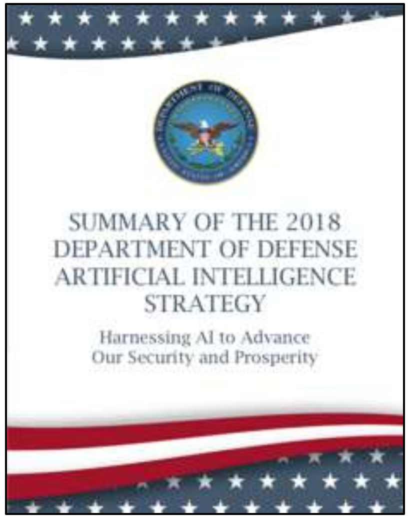
MIT Lincoln Lab Recent Advances in AI for National Security Workshop

NOVEMBER 2021

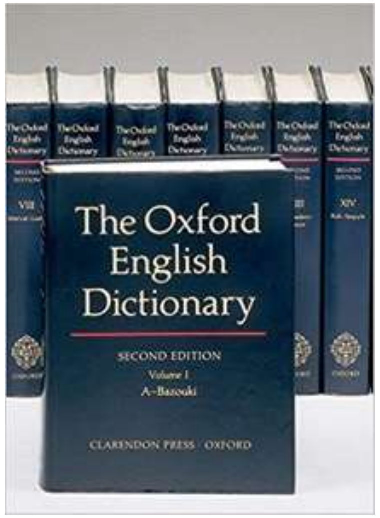
Dr. Matt Gaston
megaston@sei.cmu.edu

Director, SEI AI Division
Adjunct Associate Professor, CMU Institute for Software Research

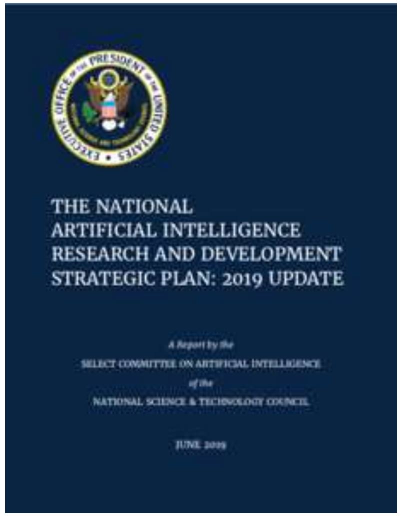
What is AI?



"AI refers to the ability of machines to perform tasks that normally require *human intelligence* – for example, recognizing patterns, learning from experience, drawing conclusions, making predictions, or taking action – whether digitally or as the smart software behind autonomous physical systems."



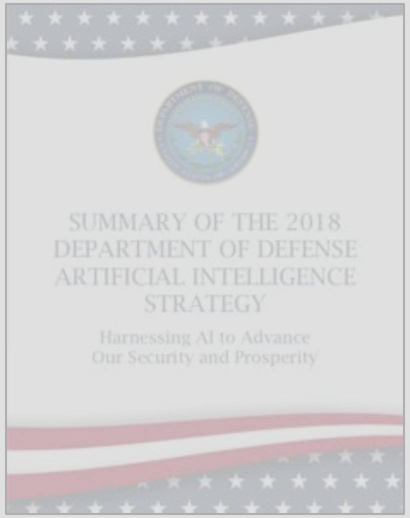
"The theory and development of computer systems able to perform tasks normally requiring *human intelligence*, such as visual perception, speech recognition, decision-making, and translation between languages."



"Artificial intelligence enables computers and other automated systems to perform tasks that have historically required *human cognition* and what we typically consider human decision-making abilities."

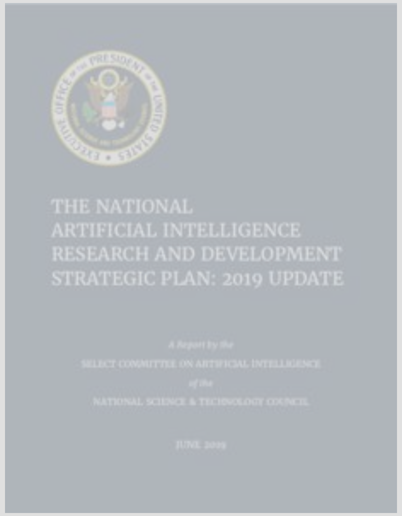
"It is the science and *engineering* of making intelligence machines, especially intelligent computer programs."

What is AI?



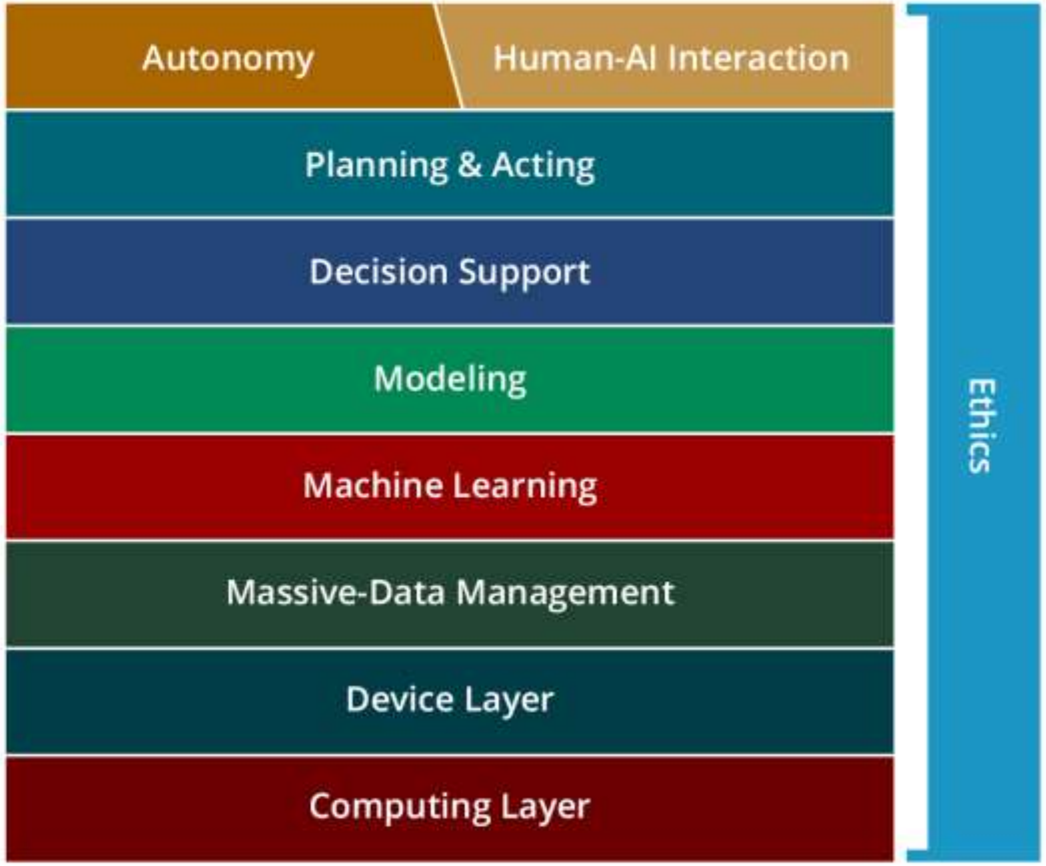
“AI refers to the ability of machines to
perform tasks that no
human intelligence –
recognizing patterns,
experience, drawing
making predictions,
whether digitally or a
software behind auto
physical systems.”

“The theory and development of
ms able to perform
requiring *human*
ch as visual
ech recognition,
g, and translation
ges.”



“Artificial intelligence
computers and other
systems to perform t
historically required *h*
and what we typically
human decision-mak

ce and *engineering* of
ence machines,
lligent computer



CMU AI Stack

Why AI Engineering?

Traditional software and system engineering are critical to building reliable AI systems, but there are important differences and gaps.

Many modern AI systems are built using machine learning.

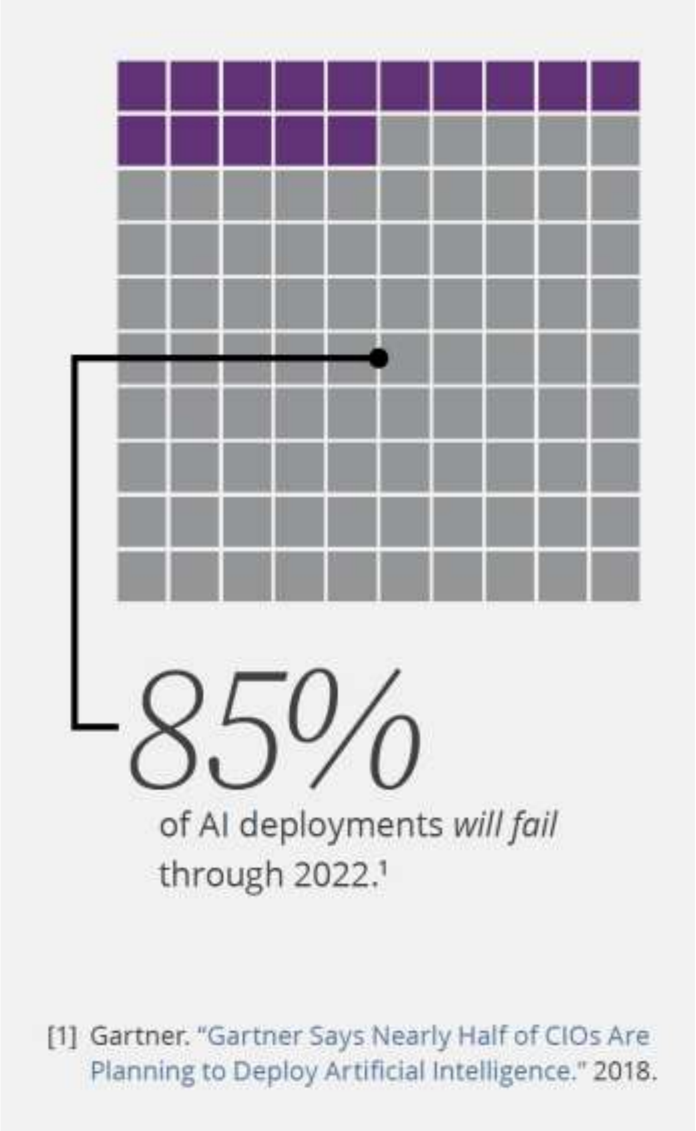
Traditional Software
- Analytical
- Explicit instructions given by programmer
- Reducible and decomposable
- Deterministic

Machine Learning
- Empirical
- Behavior learned from data or experience
- Opaque (and lots of math)
- Unpredictable

"Teaching, not micromanaging" *– Peter Norvig*

Why AI Engineering?

It is hard to get AI right.



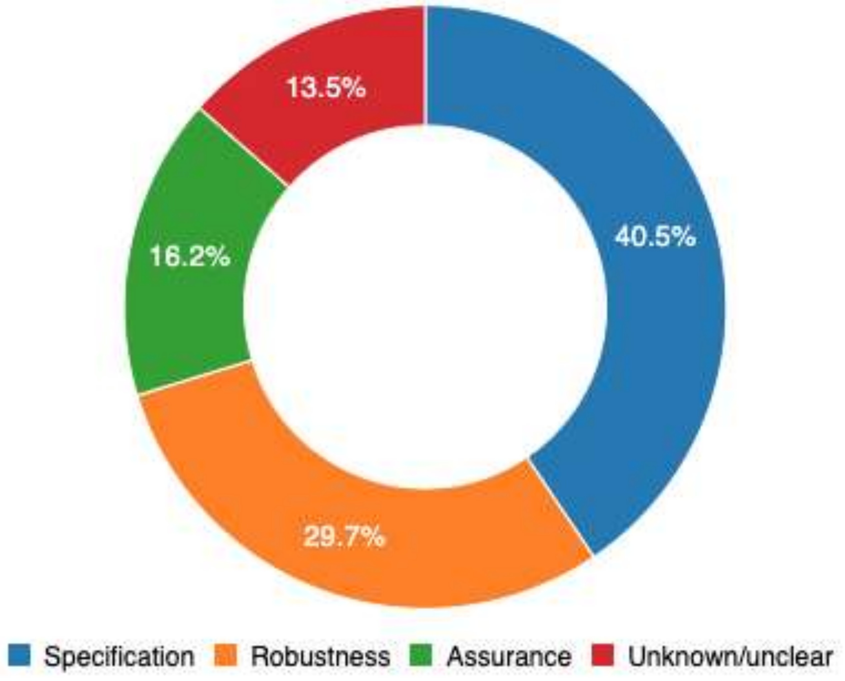
What factors cause AI system “Incidents”?

Failures in…

Specification: the system's behavior did not align with the true intentions of its designer, operator, etc.

Robustness: the system operated unsafely because of features or changes in its environment, or in the inputs the system received

Assurance: the system could not be adequately monitored or controlled during operation



74 total incidents

Source: https://incidentdatabase.ai/taxonomy/cset
Credit to Partnership on AI and the Center for Security and Emerging Technologies (CSET) at Georgetown University

Problems in ML Safety

"… we cannot rely exclusively on previous hardware and software engineering practices to create safe ML systems."

Unsolved Problems in ML Safety

Dan Hendrycks UC Berkeley
Nicholas Carlini Google
John Schulman OpenAI
Jacob Steinhardt UC Berkeley

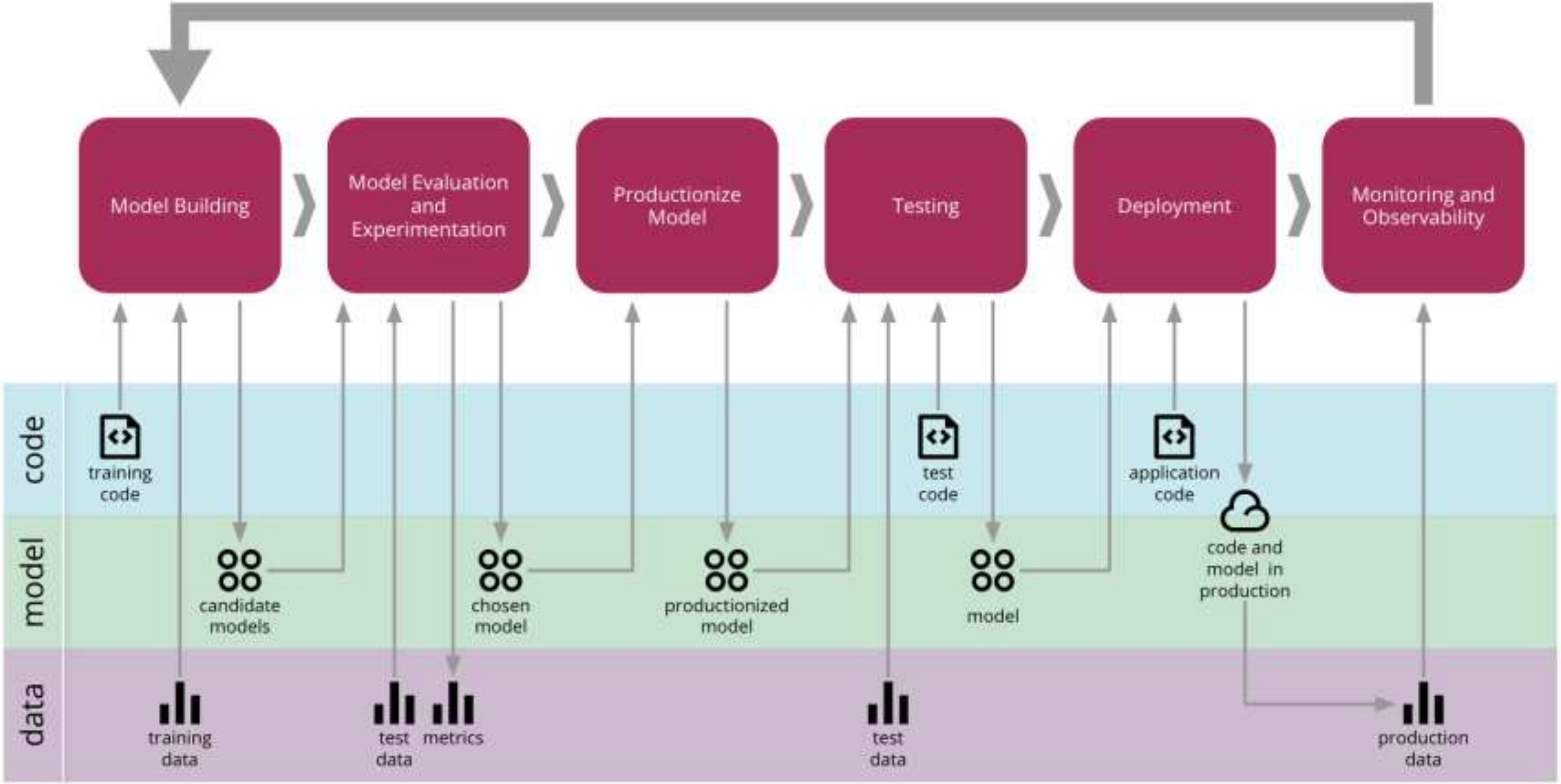
Abstract

Machine learning (ML) systems are rapidly increasing in size, are acquiring new capabilities, and are increasingly deployed in high-stakes settings. As with other powerful technologies, safety for ML should be a leading research priority. In response to emerging safety challenges in ML, such as those introduced by recent large-scale models, we provide a new roadmap for ML Safety and refine the technical problems that the field needs to address. We present four problems ready for research, namely withstanding hazards ("Robustness"), identifying hazards ("Monitoring"), steering ML systems ("Alignment"), and reducing risks to how ML systems are handled ("External Safety"). Throughout, we clarify each problem's motivation and provide concrete research directions.

Robustness	Create models that are resilient to adversaries, unusual situations, and Black Swan events.
Monitoring	Detect malicious use, monitor predictions, and discover unexpected model functionality.
Alignment	Build models that represent and safely optimize hard-to-specify human values.
External Safety	Use ML to address risks to how ML systems are handled, such as cyberattacks.

28 September 2021
https://arxiv.org/abs/2109.13916

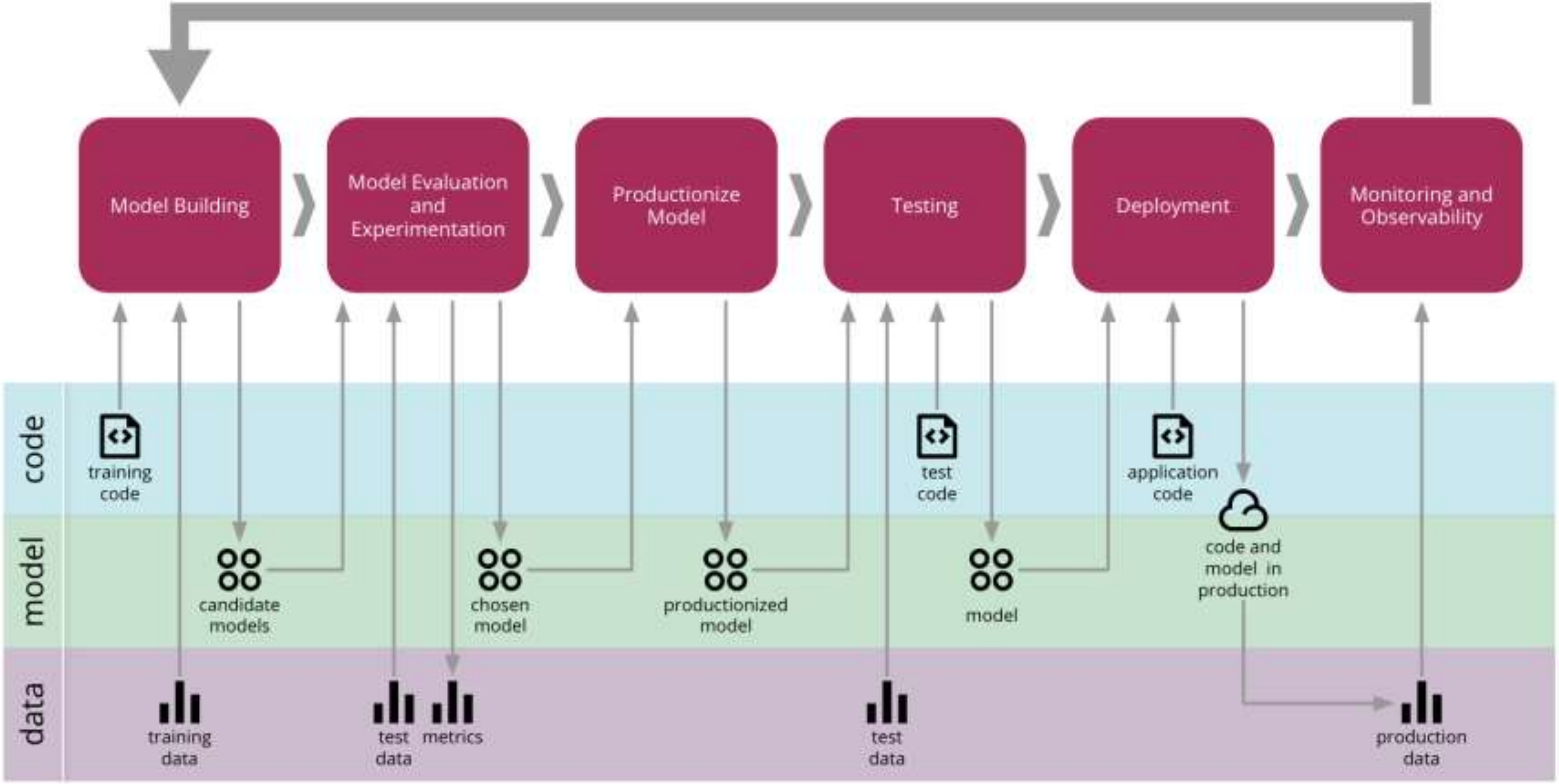
Getting ML Models into Production



Source: [Continuous Delivery for Machine Learning, Martin Fowler]

Getting ML Models into Production

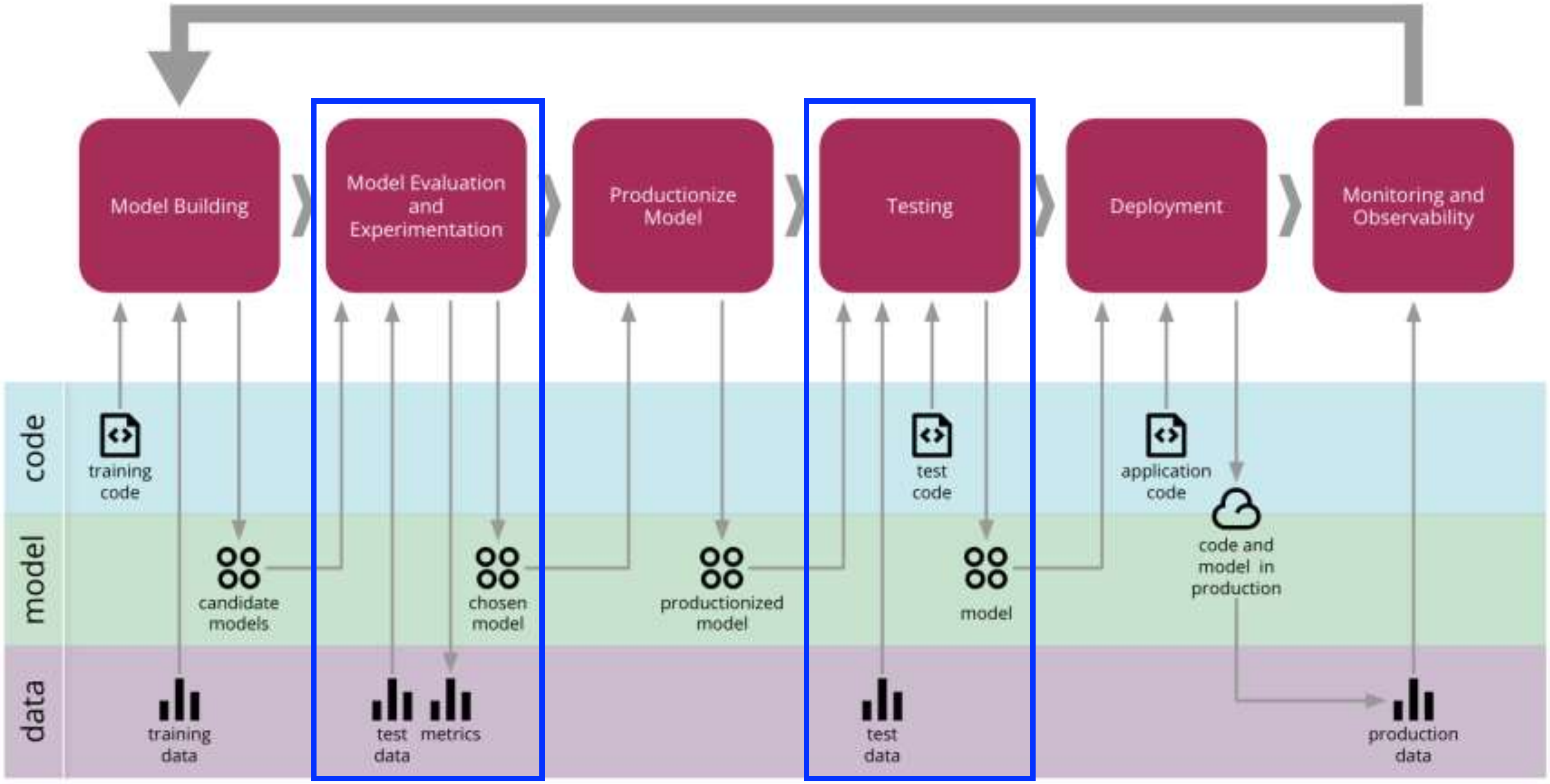
Where is Test and Evaluation, Verification and Validation (TEV&V)?



Source: [Continuous Delivery for Machine Learning, Martin Fowler]

Getting ML Models into Production

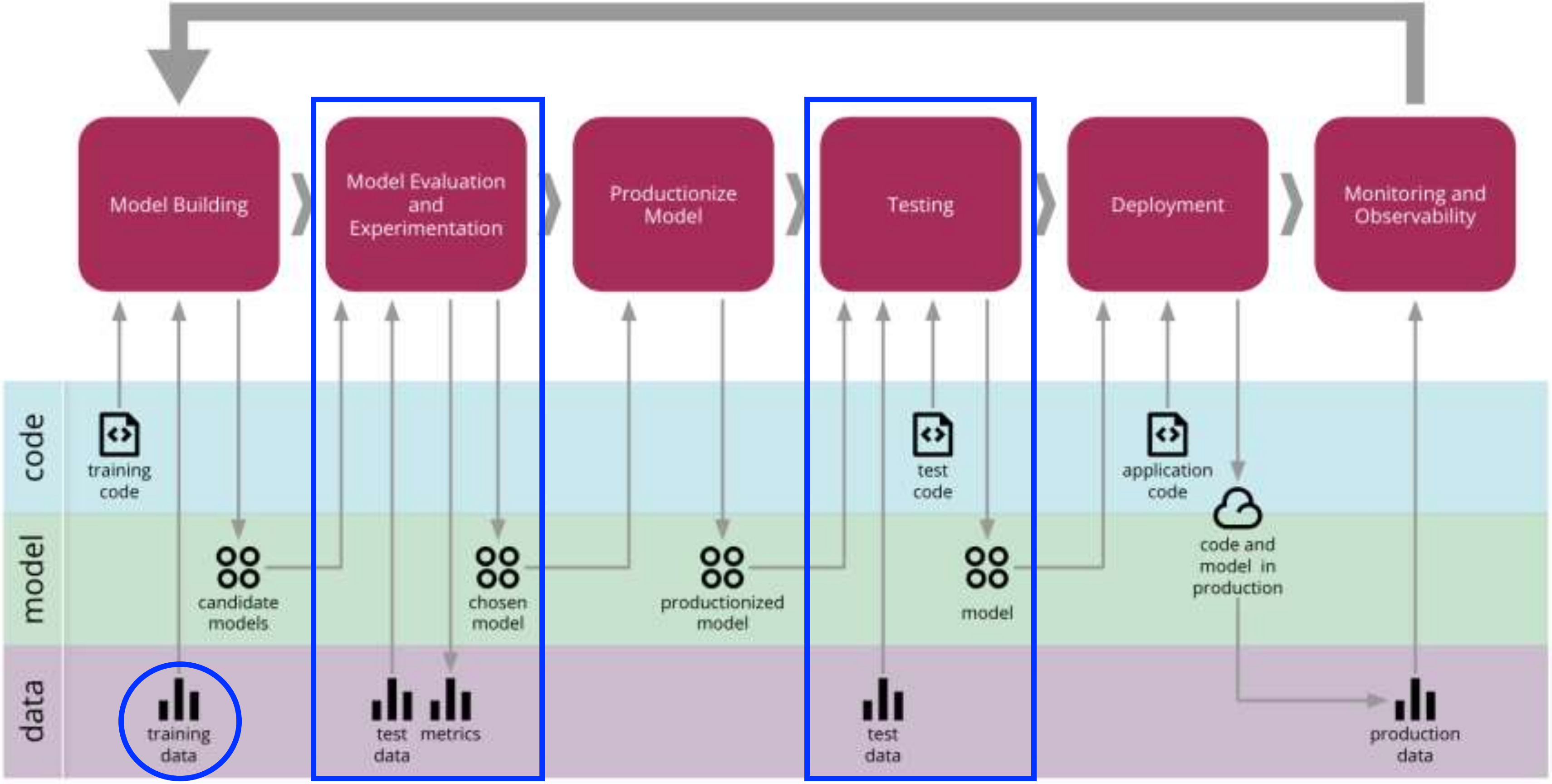
Where is Test and Evaluation, Verification and Validation (TEV&V)?



Source: [Continuous Delivery for Machine Learning, Martin Fowler]

Getting ML Models into Production

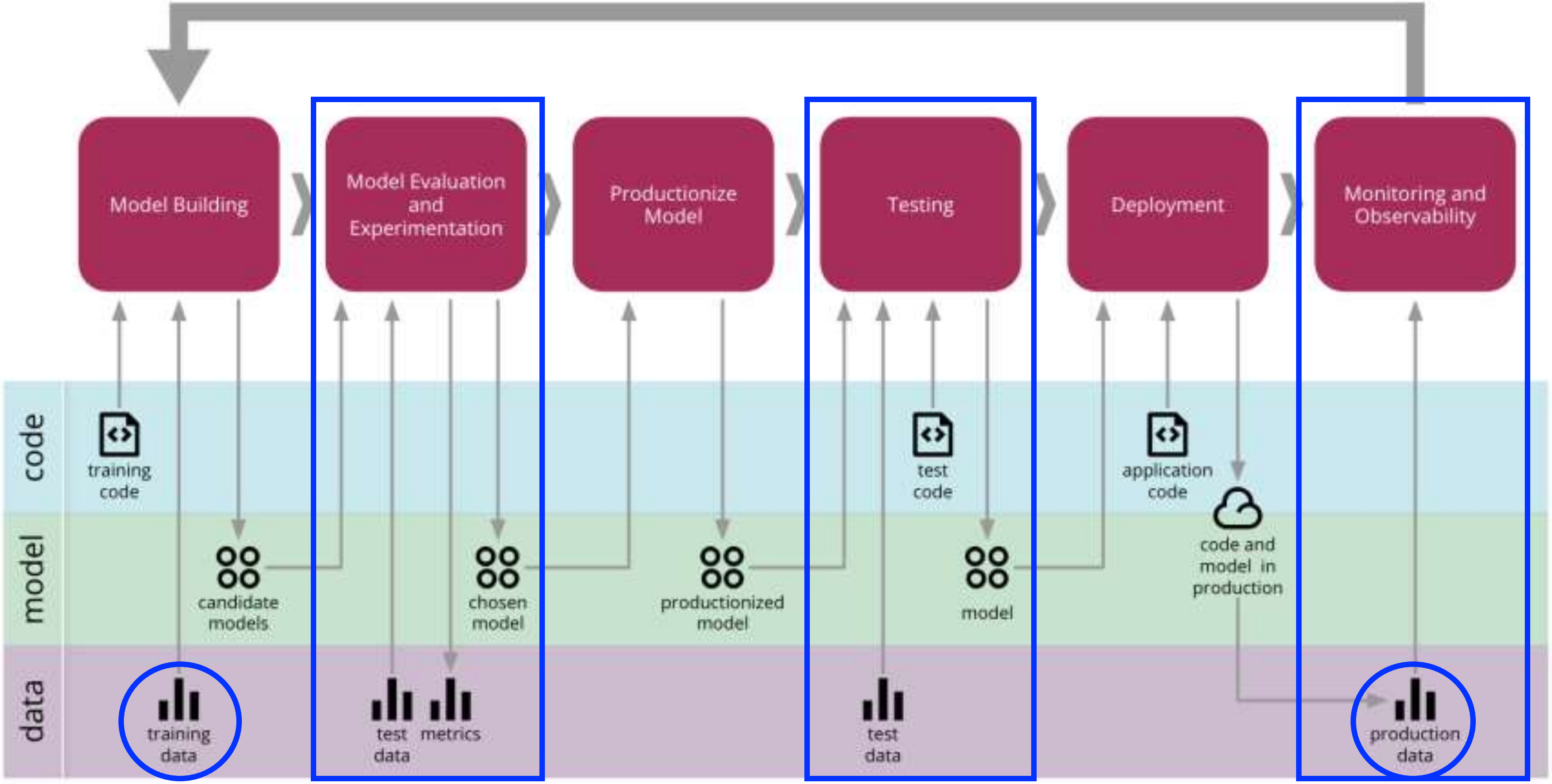
Where is Test and Evaluation, Verification and Validation (TEV&V)?



Source: [Continuous Delivery for Machine Learning, Martin Fowler]

Getting ML Models into Production

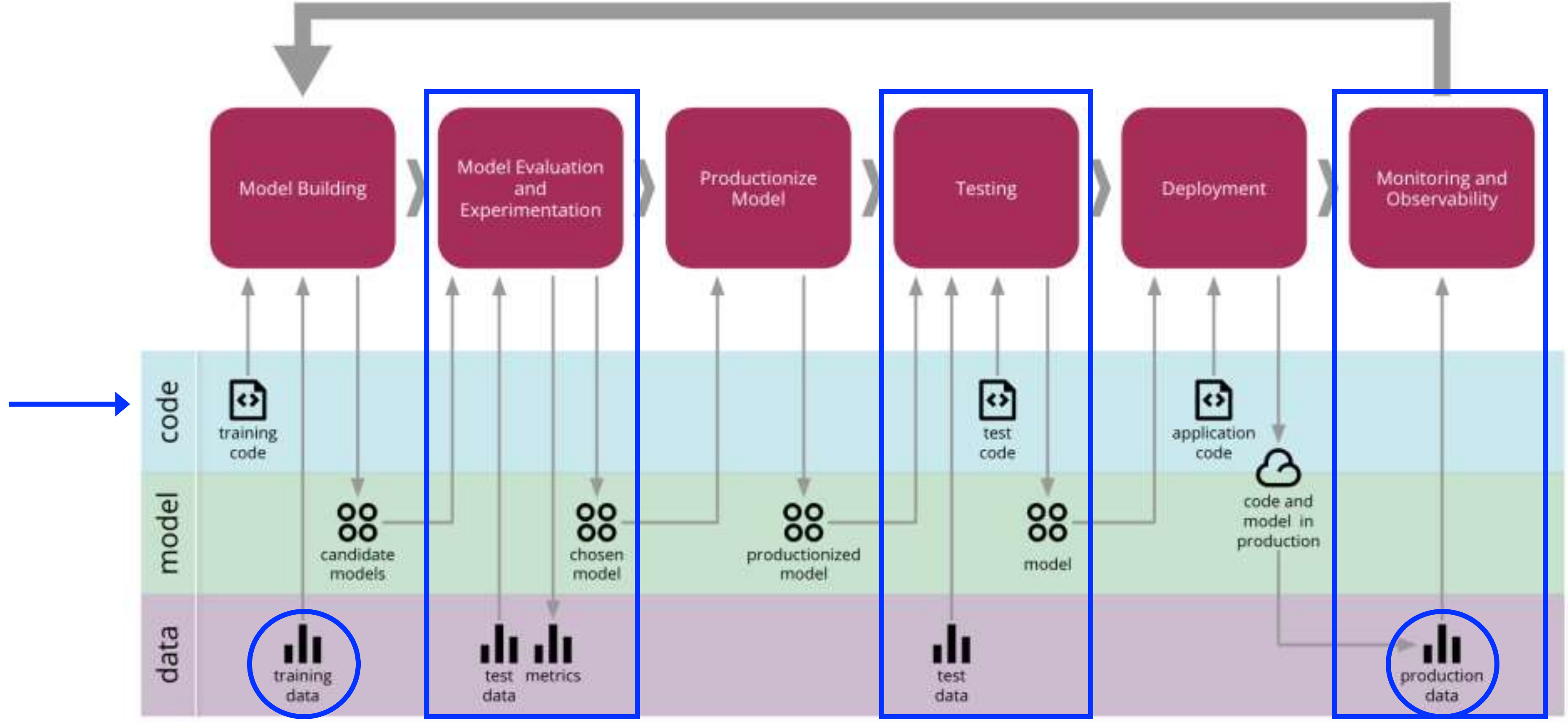
Where is Test and Evaluation, Verification and Validation (TEV&V)?



Source: [Continuous Delivery for Machine Learning, Martin Fowler]

Getting ML Models into Production

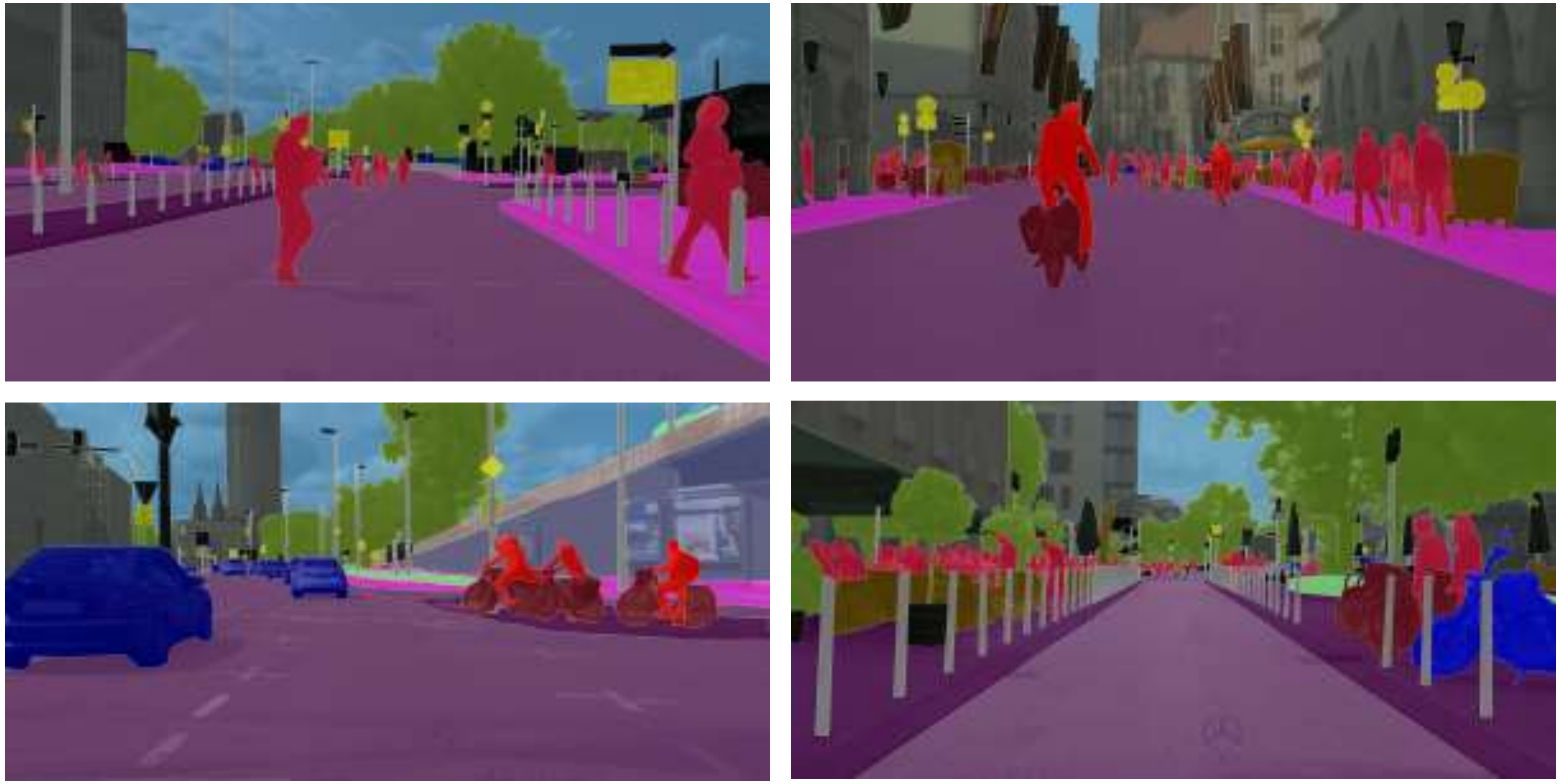
Where is Test and Evaluation, Verification and Validation (TEV&V)?



Source: [Continuous Delivery for Machine Learning, Martin Fowler]

Moving **Beyond Accuracy**: An Example

Image Segmentation



Source: https://www.cityscapes-dataset.com/

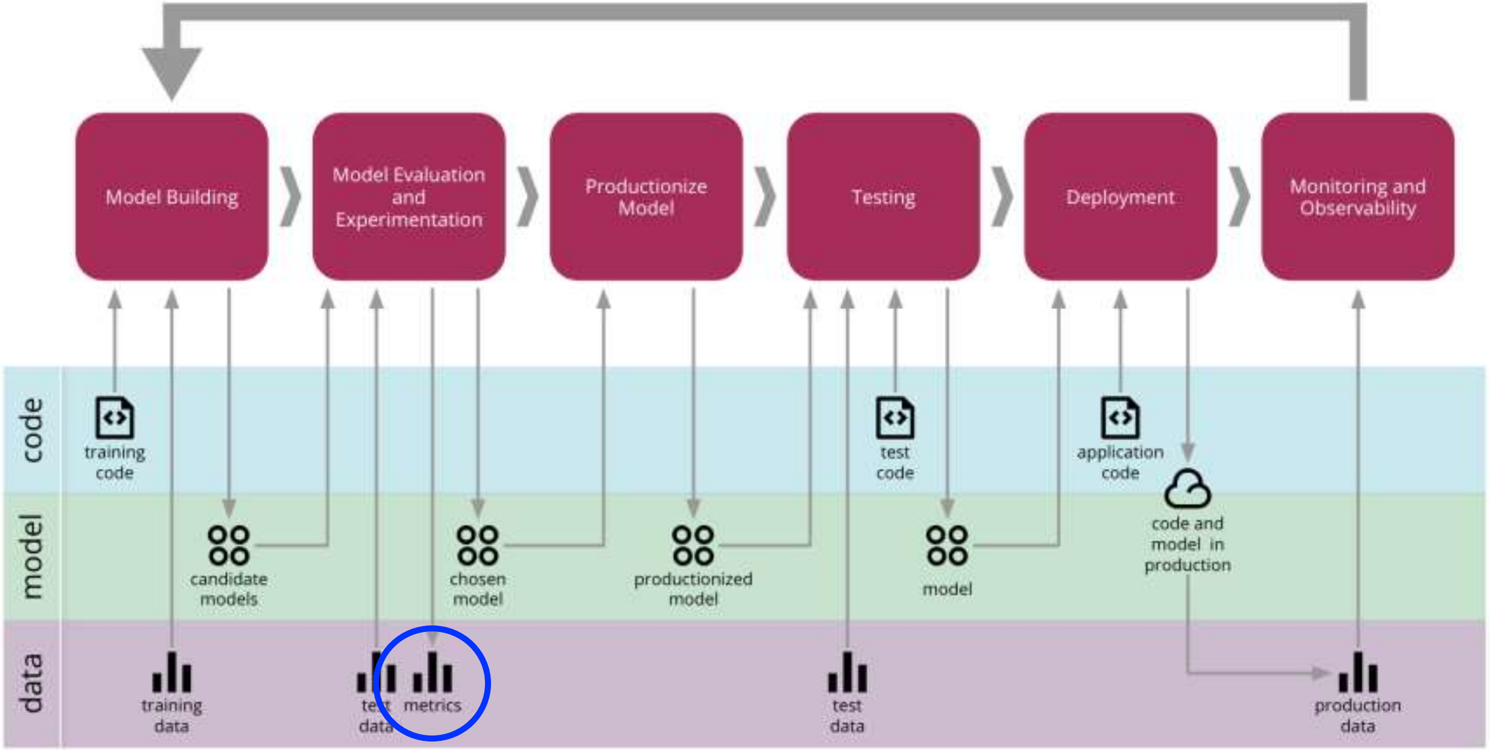
Metrics:

Pixel Accuracy (pixAcc)

Intersection over Union (IoU)

Motivating use case courtesy of:
Martial Hebert
Dean, School of Computer Science
Carnegie Mellon University

Moving **Beyond Accuracy**



Source: [Continuous Delivery for Machine Learning, Martin Fowler]

Need to understand the **tradespace** of:

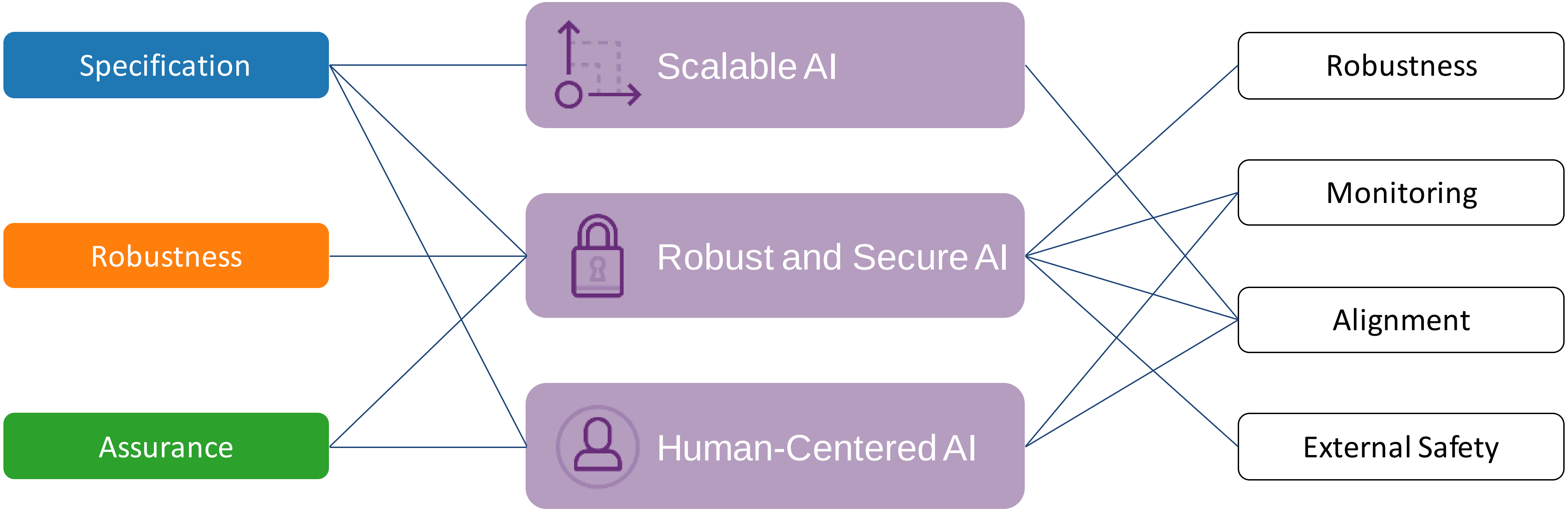
- Task accuracy
- Business/mission case
- Robustness
- Computational cost of training
- Computational cost of inference
- Deployment form factor (CSWaP)
- Risk/threat/resilience
- Interpretability/explainability
- …

AI Engineering Pillars

Scalable AI *Accommodate the size, speed, and complexity of mission needs*	• Scalable management of data and models • Enterprise scalability of AI development and deployment • Scalable algorithms and infrastructure
Robust and Secure AI *Operate reliably when faced with uncertainty or threat*	• Robustness of AI components and systems • Designing for security challenges in modern AI systems • Testing, evaluating, and analyzing AI systems
Human-Centered AI *Designed with the goal of working with, and for, people*	• Understand context of use, sense changes over time • Scope and facilitate human-machine teaming • Methods, mechanisms, and mindsets for critical oversight

Based on 2019 AI Engineering for Defense and National Security Workshop

Mapping the needs for AI Engineering



Partnership on AI & CSET

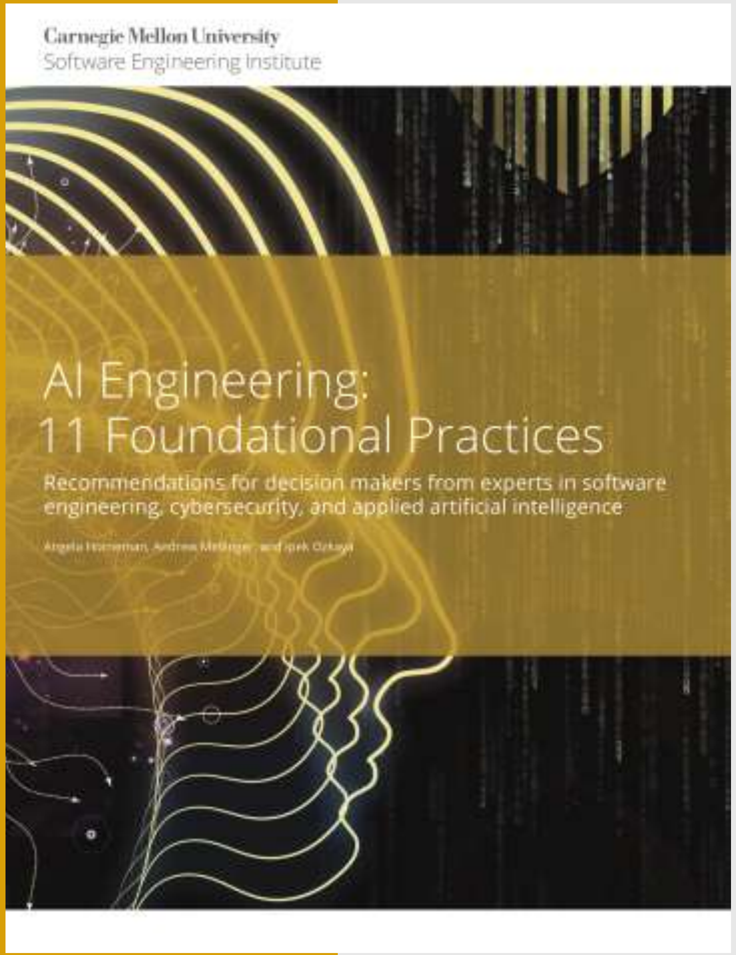
AI Engineering Pillars

Hendrycks et al.

THE NATIONAL SECURITY COMMISSION
ON ARTIFICIAL INTELLIGENCE

Chapter 7: Establishing Justified Confidence in AI Systems

1. Robust and Reliable AI
2. Human-AI Integration and Teaming
3. Test and Evaluation, Verification and Validation
4. Leadership
5. Accountability and Governance



Download Today

resources.sei.cmu.edu/library/asset-view.cfm?assetid=633647

For more information, write to **info@sei.cmu.edu**

Available for Download Today

AI Engineering: 11 Foundational Practices

"Developing viable and trusted AI systems that are deployed to the field and can be expanded and evolved for decades requires significant planning and ongoing resource commitment."

Authors

Angela Horneman, Analysis Team Lead
Carnegie Mellon University Software Engineering Institute

Andrew Mellinger, Sr. Software Developer
Carnegie Mellon University Software Engineering Institute

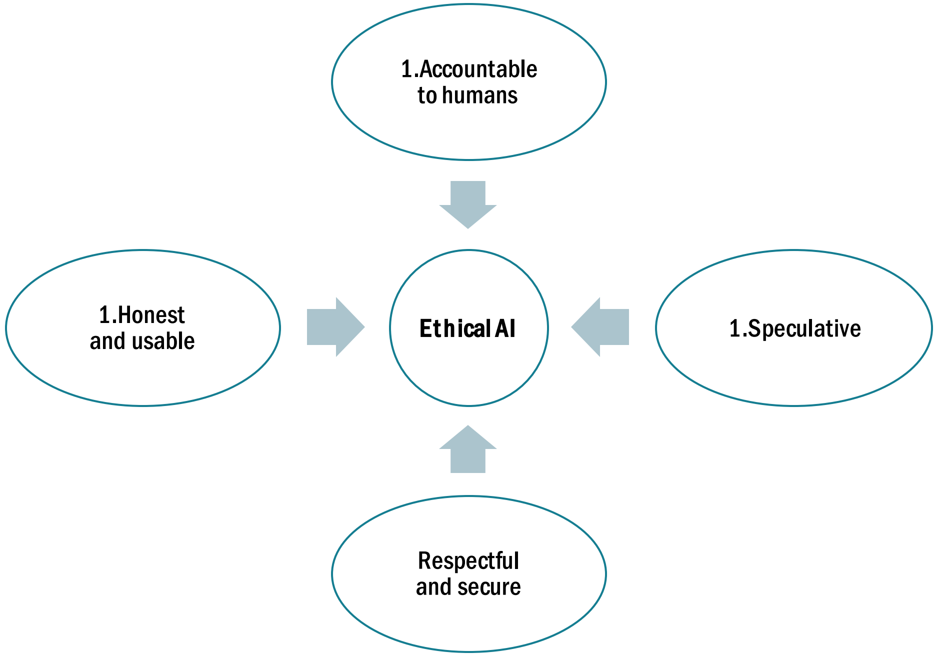
Ipek Ozkaya, Principal Researcher
Carnegie Mellon University Software Engineering Institute

Human-Centered AI

Pair Checklist with Ethical Principles

Reduce risk and unwanted bias

Support inspection and mitigation planning



Carnegie Mellon University
Software Engineering Institute

Designing Ethical AI Experiences: Checklist and Agreement

USE THIS DOCUMENT TO GUIDE THE DEVELOPMENT of accountable, de-risked, respectful, secure, honest, and usable artificial intelligence (AI) systems with a diverse team aligned on shared ethics. An initial version of this document was presented with the paper *Designing Trustworthy AI: A Human-Machine Teaming Framework to Guide Development* by Carol Smith, available at https://arxiv.org/abs/1910.03515.

We will design our AI system with the following in mind:

- ☐ Designated humans have the ultimate responsibility for all decisions and outcomes:
  - Responsibilities are explicitly defined between the AI system and human(s), and how they are shared.
  - Human responsibility will be preserved for final decisions that affect a person's life, quality of life, health, or reputation.
  - Humans are always able to monitor, control, and deactivate systems.
- ☐ Significant decisions made by the AI system will be
  - explained
  - able to be overridden
  - appealable and reversible

We work to speculatively identify the full range of risks and benefits:

- ☐ Harmful, malicious use and consequences, as well as good, beneficial use and consequences.
- ☐ We will be cognizant and exhaustively research unintended consequences.

We will create plans for the misuse/abuse of the AI system, including the following:

- ☐ communication plans to share pertinent information with all affected people
- ☐ mitigation plans for managing the identified speculative risks

We value respect and security:

- ☐ incorporating our values of humanity, ethics, equity, fairness, accessibility, diversity, and inclusion
- ☐ respecting privacy and data rights (Only necessary data will be collected.)
- ☐ providing understandable security methods
- ☐ making the AI system robust, valid, and reliable

We value transparency with the goal of engendering trust:

- ☐ The purpose, limitations, and biases of the AI system are explained in plain language.
- ☐ Data sources have unambiguous respected sources, and biases are known and explicitly stated.
- ☐ Algorithms and models are appropriate and verifiable.
- ☐ Confidence and context are presented for humans to base decisions on.
- ☐ Transparent justification for recommendations and outcomes is provided.
- ☐ Straightforward and interpretable monitoring systems are provided.

We value honesty and usability:

- ☐ Humans can easily discern when they are interacting with the AI system vs. a human.
- ☐ Humans can easily discern when and why the AI system is taking action and/or making decisions.
- ☐ Improvements will be made regularly to meet human needs and technical standards.

Team Signatures and Date

About the SEI

Contact Us

Checklist and Agreement - Downloadable PDF:
https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=636620

Human-Centered AI

Authors: Jared Dunnmon (DIU), Bryce Goodman (DIU), Peter Kirechu (DIU), Carol Smith (CMU/SEI), Alex Van Deusen (CMU/SEI)

https://www.diu.mil/responsible-ai-guidelines

Phase 1:
Planning

Phase 2:
Development

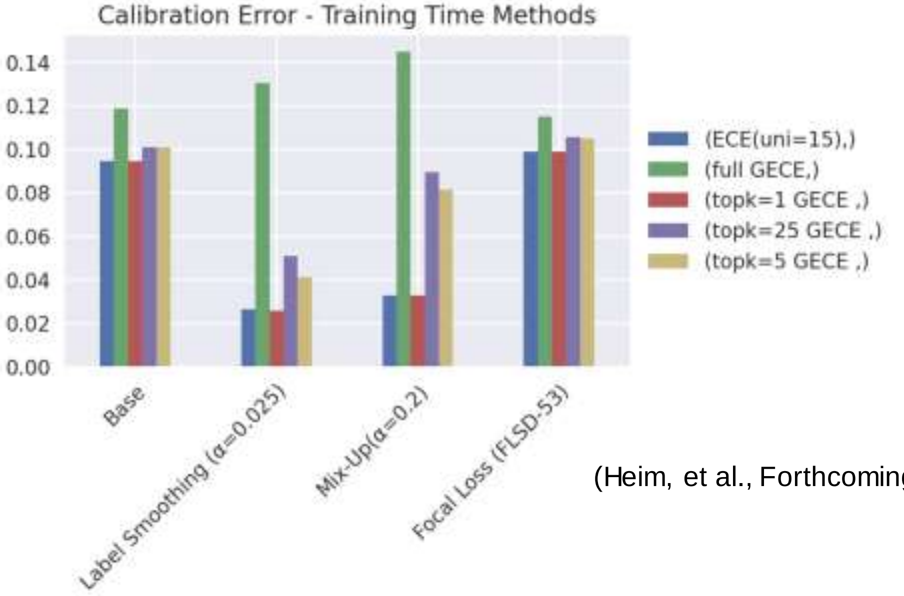
Phase 3:
Deployment

Robust (and Human-Centered) AI

Real-world data

Training Set | Data in the Wild

Classifier Calibration: The ability for a classifier to output confidences that reflect the likelihood of correct class prediction.



(Heim, et al., Forthcoming)

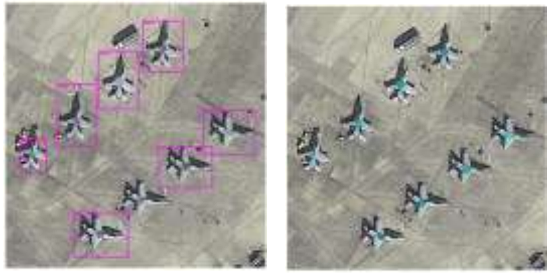
The right metrics provide tools to evaluate classifier calibration in ways that more closely represent use case deployment.

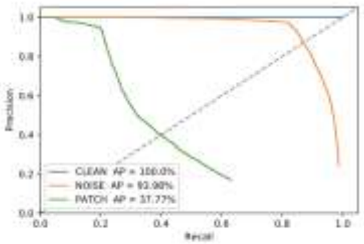
Secure AI

Learn the wrong thing

(Gu et al., 2017)

Do the wrong thing



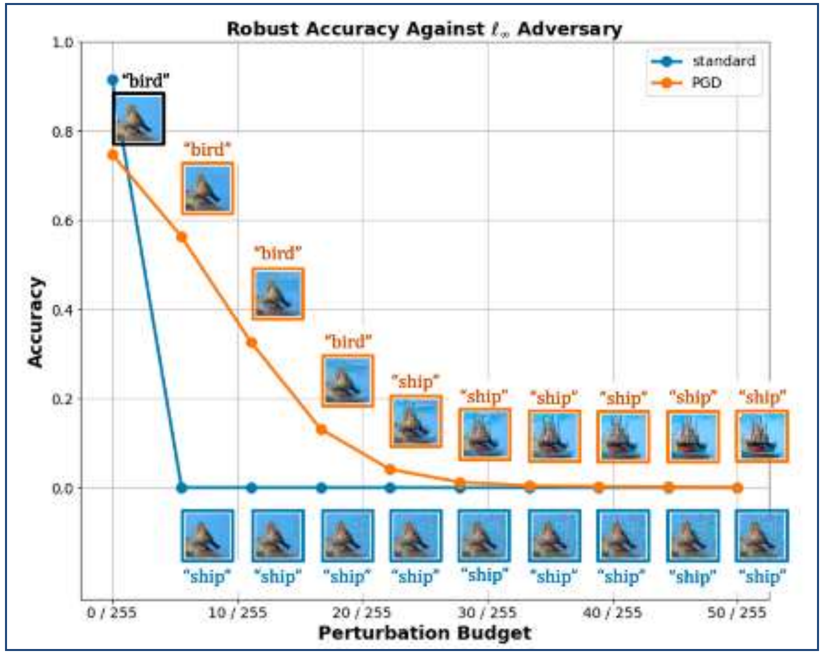


(Adhikari et al., 2020)

Reveal the wrong thing

(Fredrickson et al., 2015)

Train / Verify	Learn	Do	Reveal
Learn			
Do			
Reveal			



(VanHoudnos, et al., 2020)

Robust (and Scalable) AI

Carnegie Mellon University
Software Engineering Institute

Juneberry

Automated Framework for Training, Comparing, and Evaluating Machine Learning Models

EVALUATING MACHINE LEARNING MODELS IS CHALLENGING. Juneberry automates the training, evaluation, and comparison of multiple ML models against multiple datasets. This makes the process of verifying and validating ML models more consistent and rigorous, which reduces errors, improves reproducibility, and facilitates integration.

Why use Juneberry?

Machine learning (ML) is increasingly being applied to cybersecurity, logistics, threat detection and analysis, and other critical, data-intensive operations. To successfully adopt ML, developers must evaluate and compare the performance of ML models that may have different architectures, hyperparameters, and training data pipelines. This is not a simple task. For a true comparison, ML models must be trained and tested on the same datasets. Since the order of data affects a model's learned behavior, every training and testing dataset must be presented identically to every model. Evaluation criteria must also be consistent across models.

JUNEBERRY PROVIDES A FRAMEWORK FOR CONSISTENTLY TRAINING, EVALUATING, AND COMPARING THE PERFORMANCE OF ML MODELS. It automates loading and preparing training data, constructing and executing models, generating inferences from test datasets, producing reports, and organizing and managing different types of output.

With Juneberry, ML developers can set up structured experiments that directly compare the performance of multiple ML models with different backends. Developers define the metrics for training and evaluating these ML models. This rigorous approach to model verification and validation reduces potential errors and makes it easier to reproduce results.

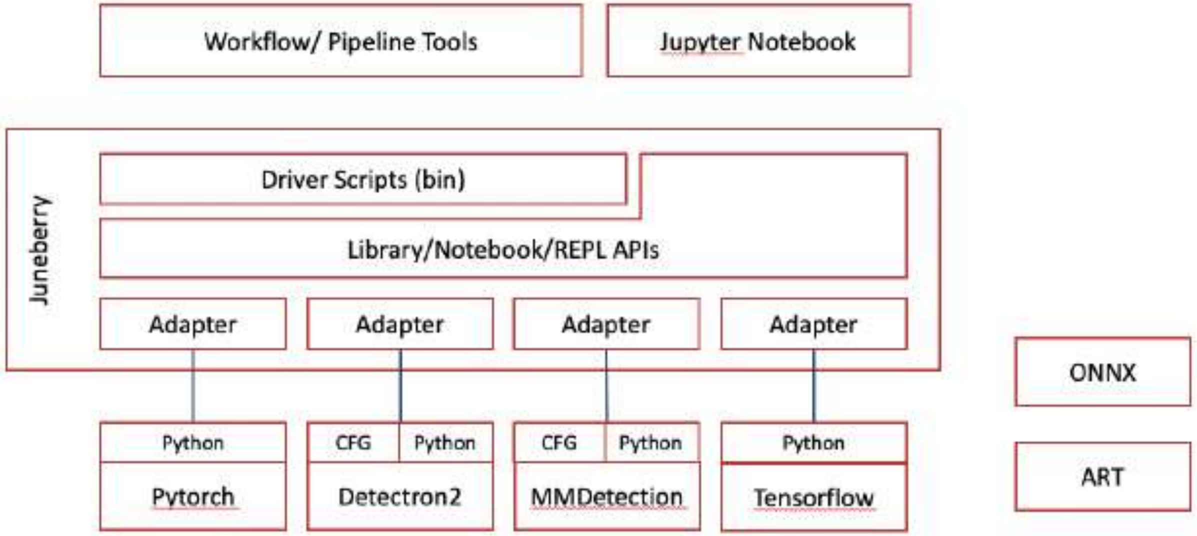
Output from Juneberry can feed into existing software development workflows. This makes it easier to integrate the best-performing ML models into applications and deploy them across the enterprise.

Key Features of Juneberry

- *Configuration-driven.* Users spend more time designing platform-independent experiments and less time writing, testing, and debugging code.
- *Supports multiple ML backends.* This provides a level playing field suitable for making comparisons between different types of ML models. PyTorch is currently supported, with support for Detectron2, MMDetection, and TensorFlow in progress.
- *Emphasizes reproducibility.* Juneberry's structured approach to training and testing allows experiments to be easily repeated and their results reproduced.

Download Juneberry:
https://github.com/cmu-sei/Juneberry

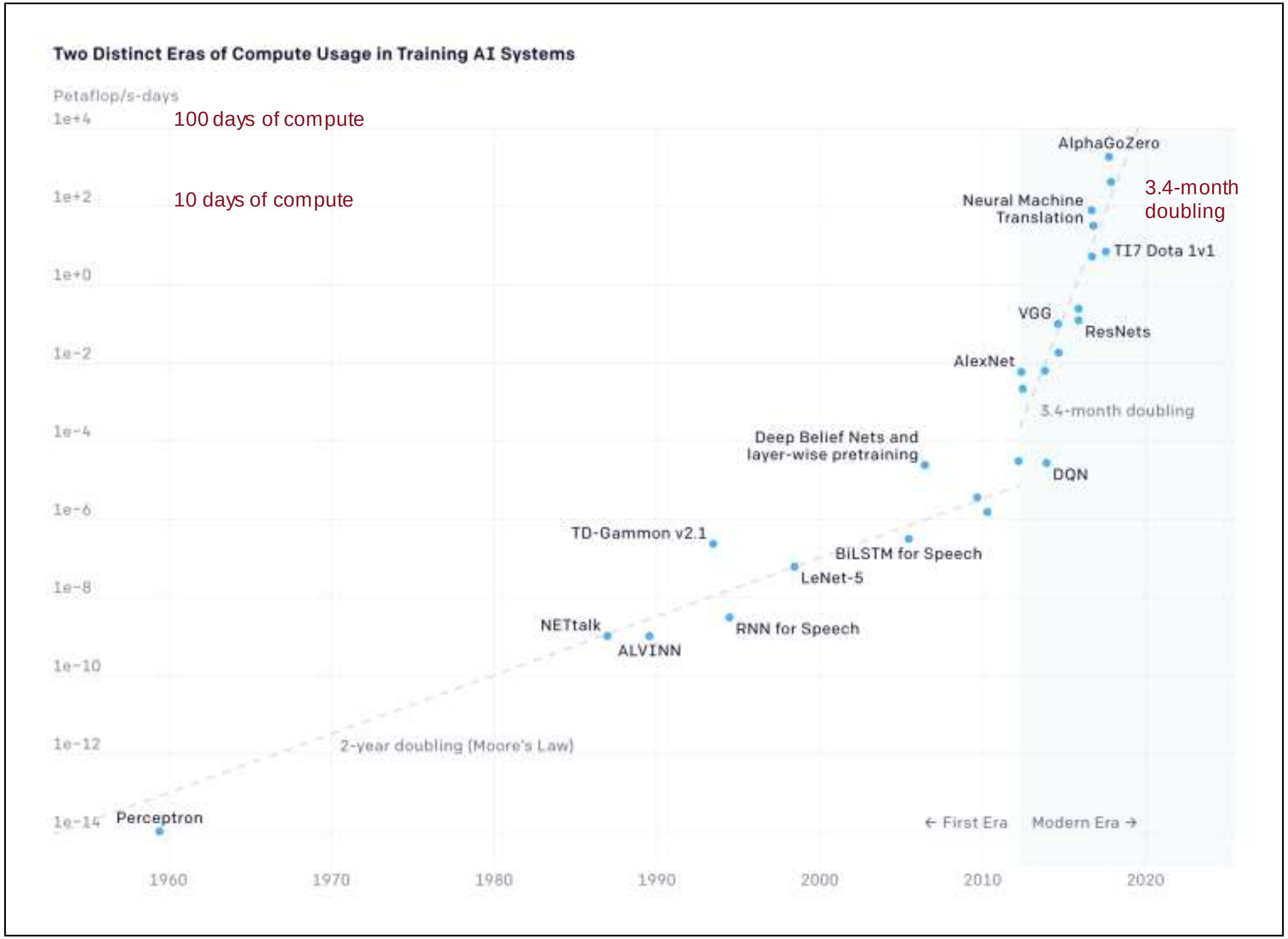
- *Evaluates and compares multiple models at a time.* The size of experiments is limited only by your computing resources.
- *Reduces the potential for error.* Automation handles the complexity of working with dozens (or even hundreds) of models and their outputs.
- *Exposes execution tasks.* This makes it easier to integrate Juneberry with existing pipelines and workflow tools such as DoIt.
- *Supports prototyping.* Juneberry is compatible with prototyping in Jupyter notebook environments.



Overview of the Juneberry framework.

Download Juneberry:
https://github.com/cmu-sei/
Juneberry

Scalable AI

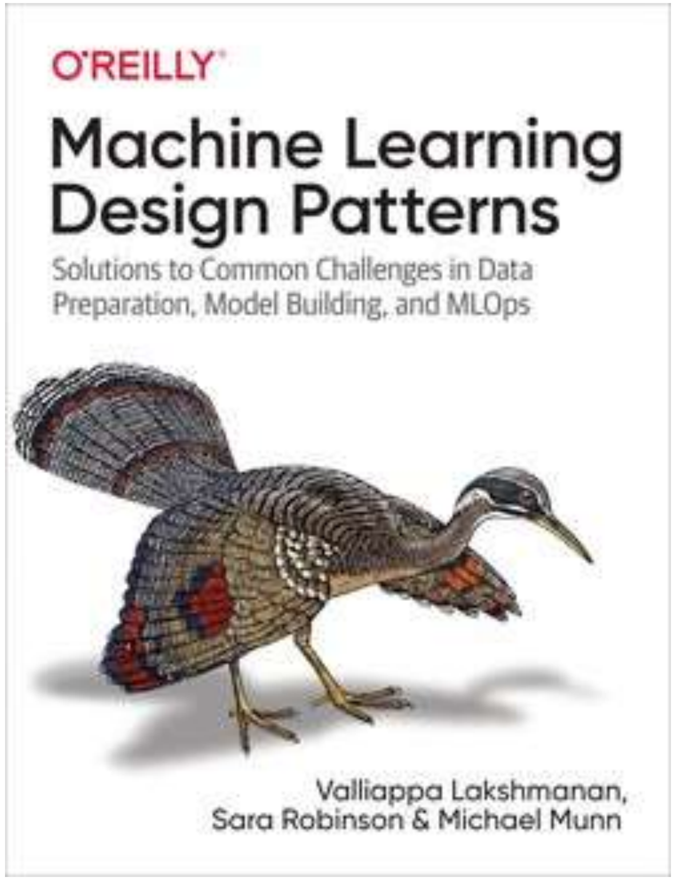



Black Hornet Nano

OpenAI: AI and Compute, May 2018. https://openai.com/blog/ai-and-compute/

Thompson et al., "The Computational Limits of Deep Learning," 2020. https://arxiv.org/pdf/2007.05558.pdf

Scalable (and Human-Centered) AI



https://www.oreilly.com/library/view/machine-learning-design

arXiv.org > cs > arXiv:2107.00079

Search... All fields Search

Help | Advanced Search

Computer Science > Machine Learning

[Submitted on 30 Jun 2021]

Using AntiPatterns to avoid MLOps Mistakes

Nikhil Muralidhar, Sathappah Muthiah, Patrick Butler, Manish Jain, Yu Yu, Katy Burne, Weipeng Li, David Jones, Prakash Arunachalam, Hays 'Skip' McCormick, Naren Ramakrishnan

We describe lessons learned from developing and deploying machine learning models at scale across the enterprise in a range of financial analytics applications. These lessons are presented in the form of antipatterns. Just as design patterns codify best software engineering practices, antipatterns provide a vocabulary to describe defective practices and methodologies. Here we catalog and document numerous antipatterns in financial ML operations (MLOps). Some antipatterns are due to technical errors, while others are due to not having sufficient knowledge of the surrounding context in which ML results are used. By providing a common vocabulary to discuss these situations, our intent is that antipatterns will support better documentation of issues, rapid communication between stakeholders, and faster resolution of problems. In addition to cataloging antipatterns, we describe solutions, best practices, and future directions toward MLOps maturity.

Subjects: Machine Learning (cs.LG)
Cite as: arXiv:2107.00079 [cs.LG]
(or arXiv:2107.00079v1 [cs.LG] for this version)

Submission history

From: Nikhil Muralidhar [view email]
[v1] Wed, 30 Jun 2021 20:00:52 UTC (906 KB)

Download:
- PDF
- Other formats

Current browse context:
cs.LG
< prev | next >
new | recent | 2107
Change to browse by:
cs

References & Citations
- NASA ADS
- Google Scholar
- Semantic Scholar

Export Bibtex Citation

Bookmark

https://arxiv.org/abs/2107.00079

https://www.sei.cmu.edu/our-work/artificial-intelligence-engineering/

Advocate for AI Engineering

Collaborate to Build the Discipline

Support the Research Agenda