The Sixteenth International Conference on Software Engineering Advances
ICSEA 2021 https://www.iaria.org/conferences2021/ICSEA21.html
Oct 3-7, 2021 - Barcelona, Spain and Online

Machine Learning Software Engineering Patterns: Classification and Practitioners' Insights

Hironori Washizaki

Professor at Waseda University, Tokyo, Japan

washizaki@waseda.jp

WASEDA University

https://www.waseda.jp/culture/news/2020/04/30/10381/

Prof. Dr. Hironori Washizaki

- Professor and the Associate Dean of the Research Promotion Division at Waseda University in Tokyo
- Visiting Professor at the National Institute of Informatics
- Outside Directors of SYSTEM INFORMATION and eXmotion
- Research and education projects
  - Leading a large-scale grant at MEXT enPiT-Pro Smart SE
  - **Leading framework team of JST MIRAI eAI project**
- Professional contributions
  - IEEE Computer Society Vice President for Professional and Educational Activities
  - Editorial Board Member of MDPI Education Sciences
  - Steering Committee Member of the IEEE Conference on Software Engineering Education and Training (CSEE&T)
  - Associate Editor of IEEE Transactions on Emerging Topics in Computing
  - Advisory Committee Member of the IEEE-CS COMPSAC
  - Steering Committee Member of Asia-Pacific Software Engineering Conference (APSEC)
  - Convener of ISO/IEC/JTC1 SC7/WG20

Machine Learning Software Engineering Patterns: Classification and Practitioners' Insights

Hironori Washizaki

Professor at Waseda University, Tokyo, Japan

washizaki@waseda.jp http://www.washi.cs.waseda.ac.jp/

- Hironori Washizaki, Hiromu Uchida, Foutse Khomh and Yann-Gael Gueheneuc, "Studying Software Engineering Patterns for Designing Machine Learning Systems," The 10th International Workshop on Empirical Software Engineering in Practice (IWESEP 2019)
- Hironori Washizaki, Foutse Khomh, Yann-Gael Gueheneuc, "Software Engineering Patterns for Machine Learning Applications (SEP4MLA)," 9th Asian Conference on Pattern Languages of Programs (AsianPLoP 2020)
- Hironori Washizaki, Foutse Khomh, Yann-Gaël Guéhéneuc, Hironori Takeuchi, Satoshi Okuda, Naotake Natori, Naohisa Shioura, "Software Engineering Patterns for Machine Learning Applications (SEP4MLA) – Part 2", the 27th Conference on Pattern Languages of Programs in 2020 (PLoP'20)
- Hironori Washizaki, Hironori Takeuchi, Foutse Khomh, Naotake Natori, Takuo Doi, Satoshi Okuda, "Practitioners' insights on machine-learning software engineering design patterns: a preliminary study," 36th IEEE International Conference on Software Maintenance and Evolution (ICSME 2020), Late Breaking Ideas track
- Yasuhiro Watanabe, Hironori Washizaki, Kazunori Sakamoto, Daisuke Saito, Kiyoshi Honda, Naohiko Tsuda, Yoshiaki Fukazawa, Nobukazu Yoshioka, "Preliminary Literature Review of Machine Learning System Development Practices," 45th IEEE Computer Society Signature Conference on Computers, Software and Applications (COMPSAC 2021), Fast Abstract
- Jati H. Husen, Hnin Thandar Tun, Nobukazu Yoshioka, Hironori Washizaki and Yoshiaki Fukazawa, "Goal-Oriented Machine Learning-Based Component Development Process," ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS 2021), Poster

Supported by JST MIRAI eAI JPMJMI20B8 https://www.jst.go.jp/mirai/jp/program/super-smart/JPMJMI20B8.html

Agenda

- ML software engineering and patterns
- Literature review of software engineering patterns for ML applications
- Classification of ML patterns
- Practitioners’ insights on ML patterns

Street Cafe

Problem: Needs to have a place where people can sit lazily, legitimately, be on view, and watch the world go by…

Solution: Encourage local cafes to spring up in each neighborhood. Make them intimate places, with several rooms, open to a busy path …

Alexander, Christopher, et al. *A Pattern Language*. Oxford University Press, 1977.

https://unsplash.com/photos/8IKf54pc3qk

https://unsplash.com/photos/zACLEreWKXE

Towards a pattern language

… OK, so, to attract many people to our city, **Small Public Squares** should be located in the center. At the **Small Public Square**, make **Street Cafes** be **Opening to the Street** …

https://unsplash.com/photos/EdpbTj3Br-Y

Small Public Square

https://unsplash.com/photos/zFoRwZirFvY

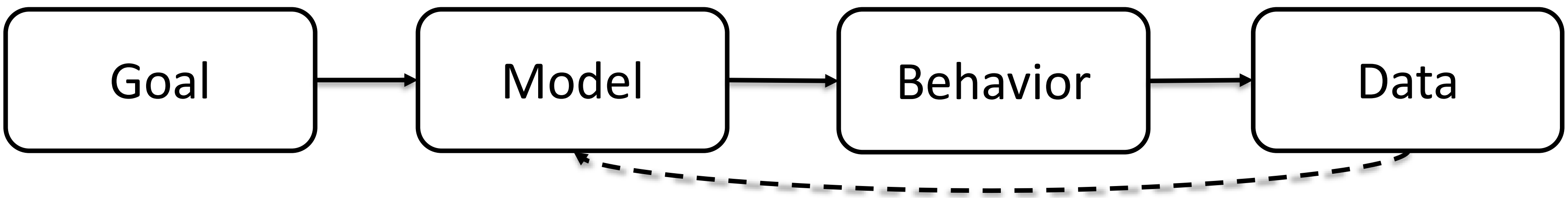
Street Cafe

https://unsplash.com/photos/GqurqYbj7aU

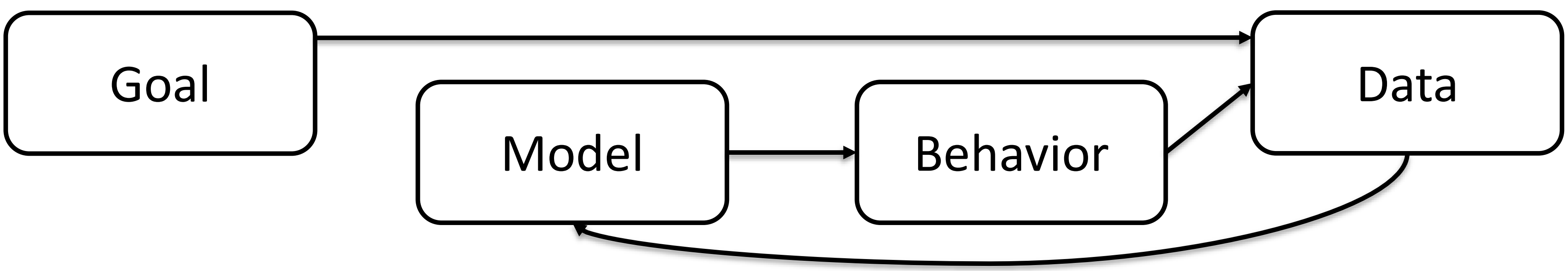
Opening to the Street

ML-SE: Induction (and abduction)

Conventional software: Deduction



ML software engineering: Induction (and abduction)

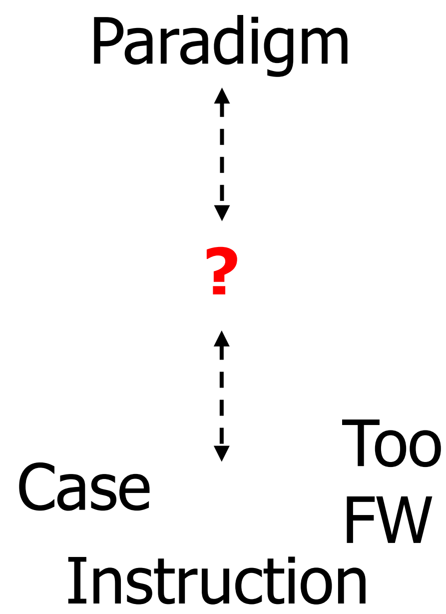


H. Maruyama, “Machine Learning Engineering and Reuse of AI Work Products,” The First International Workshop on Sharing and Reuse of AI Work Products, 2017

Hironori Washizaki, “Towards Software Value Co-Creation with AI”, The 44th IEEE Computer Society Signature Conference on Computers, Software, and Applications (COMPSAC 2020), Fast Abstract

ML software engineering needs patterns!

- **Bridge** between abstract paradigms and concrete cases/tools
  - Documenting Know-Why, Know-What and Know-How
  - Reusing solutions and problems
  - Getting consistent architecture



- **Common language** among stakeholders
  - Software engineers, data scientist, domain experts, network engineers, …

Practices and patterns in ML-SE

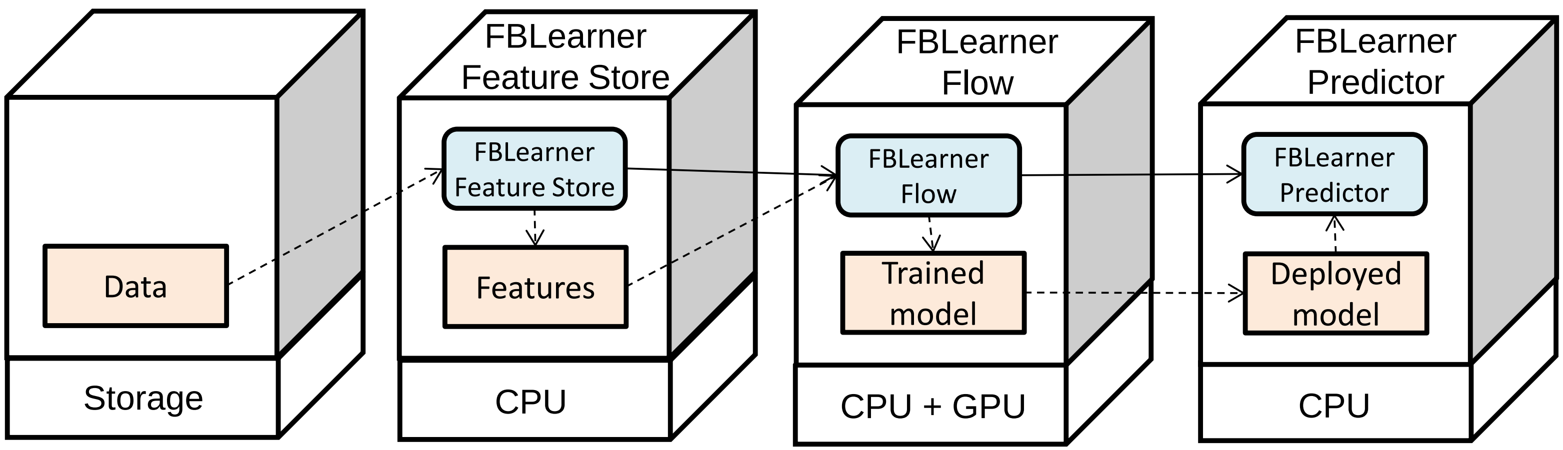
- Researchers and practitioners studying best practices strive to design Machine Learning (ML) systems and software.
- Some practices are formalized as patterns.

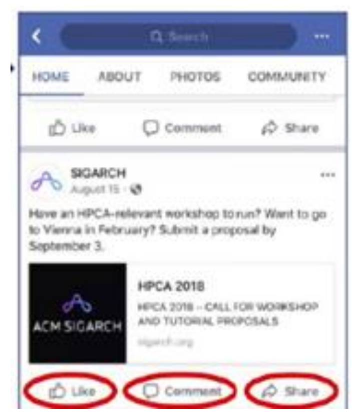
(NOTE: NOT handle ML model patterns.)

Data Lake for ML

https://unsplash.com/photos/_HPIEmsKgP0

Different Workloads in Different Computing Environments (e.g., Facebook)

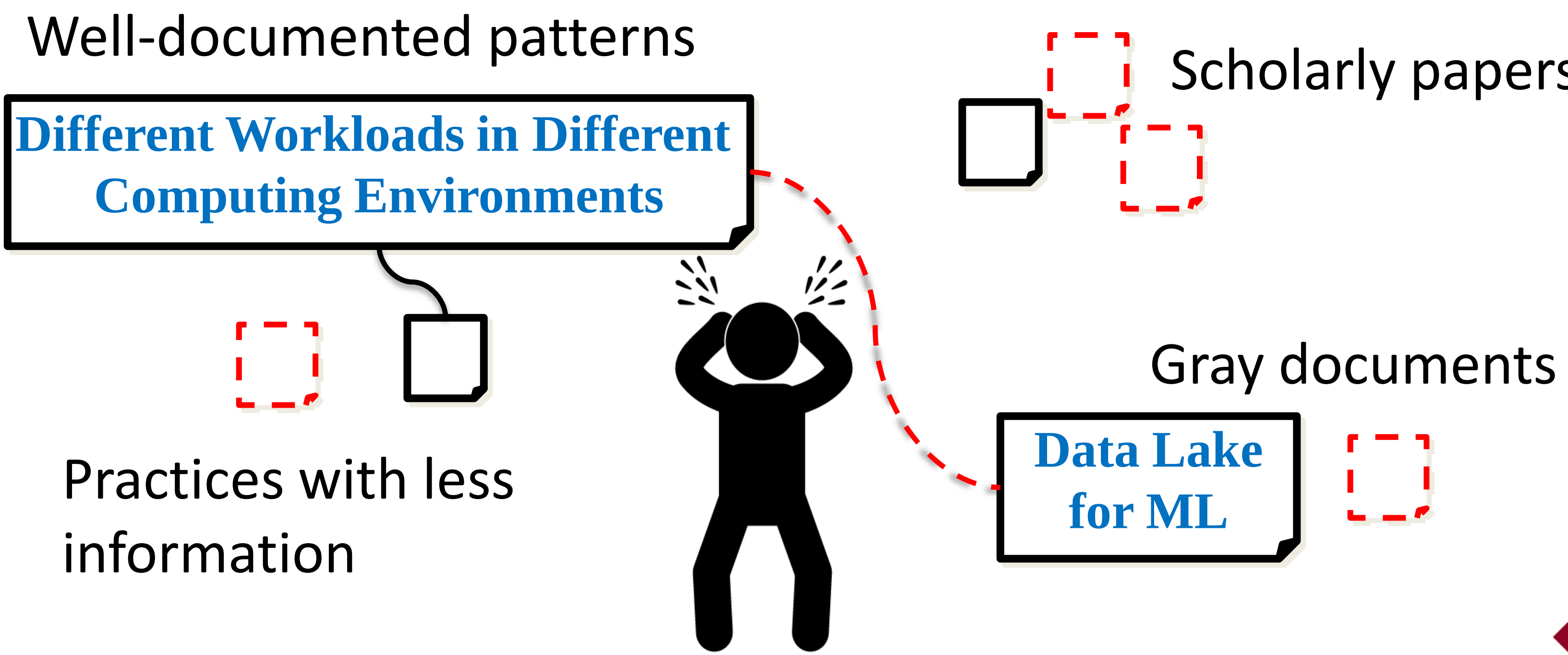




K. M. Hazelwood, et al., Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective, HPCA 2018
H. Washizaki, et al. Software Engineering Patterns for Machine Learning Applications (SEP4MLA) – Part 2, PLoP’20

Problem and goal

- ML system architecture and design patterns at different abstraction levels are not well classified and studied.
- Thus, we conducted a survey of software developers and an Systematic Literature Review.



Agenda

- ML software engineering and patterns
- Literature review of software engineering patterns for ML applications
- Classification of ML patterns
- Practitioners' insights on ML patterns

Research questions

- **RQ1. Does academic and gray literature address the design of ML systems and software?**
  - 19 scholarly and 19 gray documents identified
  - 15 SE patterns for ML applications extracted
- **RQ2. Can ML patterns be classified?**
  - Categories of scopes: Topology, programming and model
  - Quality attributes: ISO/IEC 25010:2011 System and software product quality attributes, ML model and prediction quality attributes
- **RQ3. How do practitioners perceive ML patterns?**
  - Questionnaire-based survey for 600+ developers
  - Developers were unfamiliar with most ML patterns, although there were several major patterns used by 20%

RQ1. Does academic and gray literature address the design of ML systems and software?

- Systematic Literature Review (SLR)
  - Scholar papers: Engineering Village
  - Gray documents: Google
- 19 scholarly papers and 19 gray documents identified
- 15 patterns extracted

Engineering Village

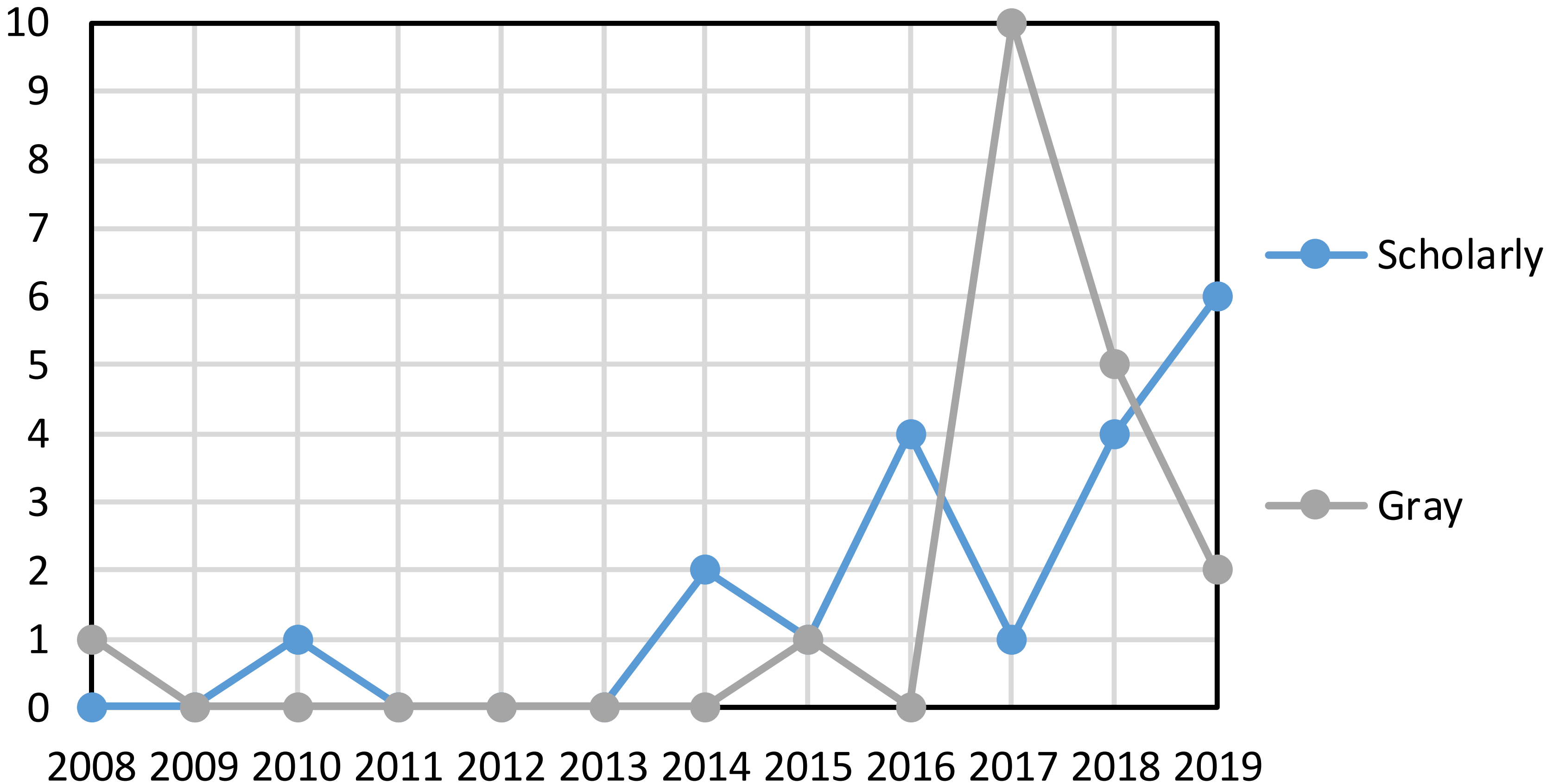
((((system) OR (software)) AND (machine learning) AND (implementation pattern) OR (pattern) OR (architecture pattern) OR (design pattern) OR (anti-pattern) OR (recipe) OR (workflow) OR (practice) OR (issue) OR (template))) WN ALL) + ((cpx OR ins OR kna) WN DB) AND (({ca} OR {ja} OR {ip} OR {ch}) WN DT)

Google

(system OR software) "Machine learning" (pattern OR "implementation pattern" OR "architecture pattern" OR "design pattern“ OR anti-pattern OR recipe OR workflow OR practice OR issue OR template)
"machine implementation pattern" OR "architecture pattern" OR "design pattern“ OR anti-pattern OR recipe OR workflow OR practice OR issue OR template

Numbers of Documents per Year

- ML application systems have recently become popular due to the promotion of artificial intelligence.
- Since 2008, academic and gray documents have discussed good (bad) practices of ML application systems design.



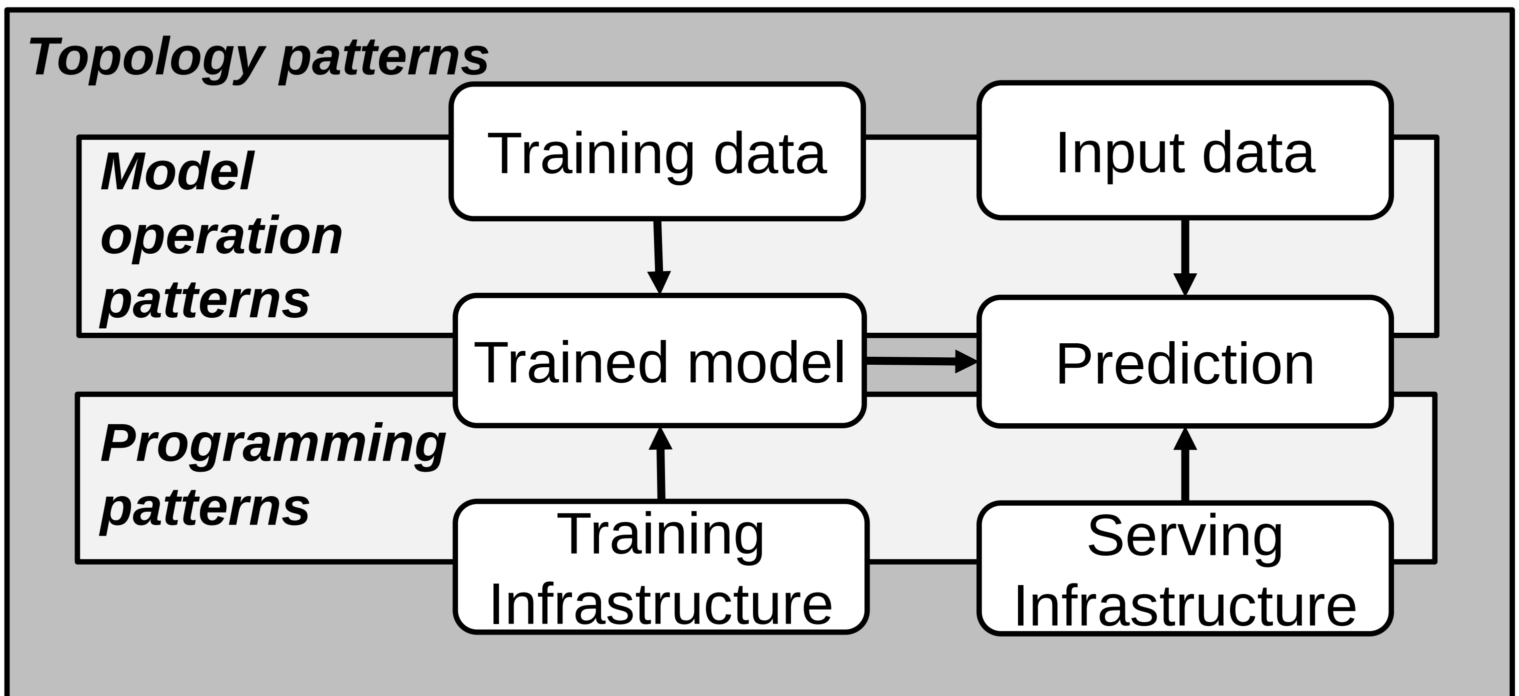
H. Washizaki, et al., Studying Software Engineering Patterns for Designing Machine Learning Systems, IWESEP 2019

Agenda

- ML software engineering and patterns
- Literature review of software engineering patterns for ML applications
- Classification of ML patterns
- Practitioners' insights on ML patterns

RQ2. Can ML patterns be classified?

- Model operation patterns that focus on ML models
- Programming patterns that define the design of a particular component
- Topology patterns that define the entire system architecture.



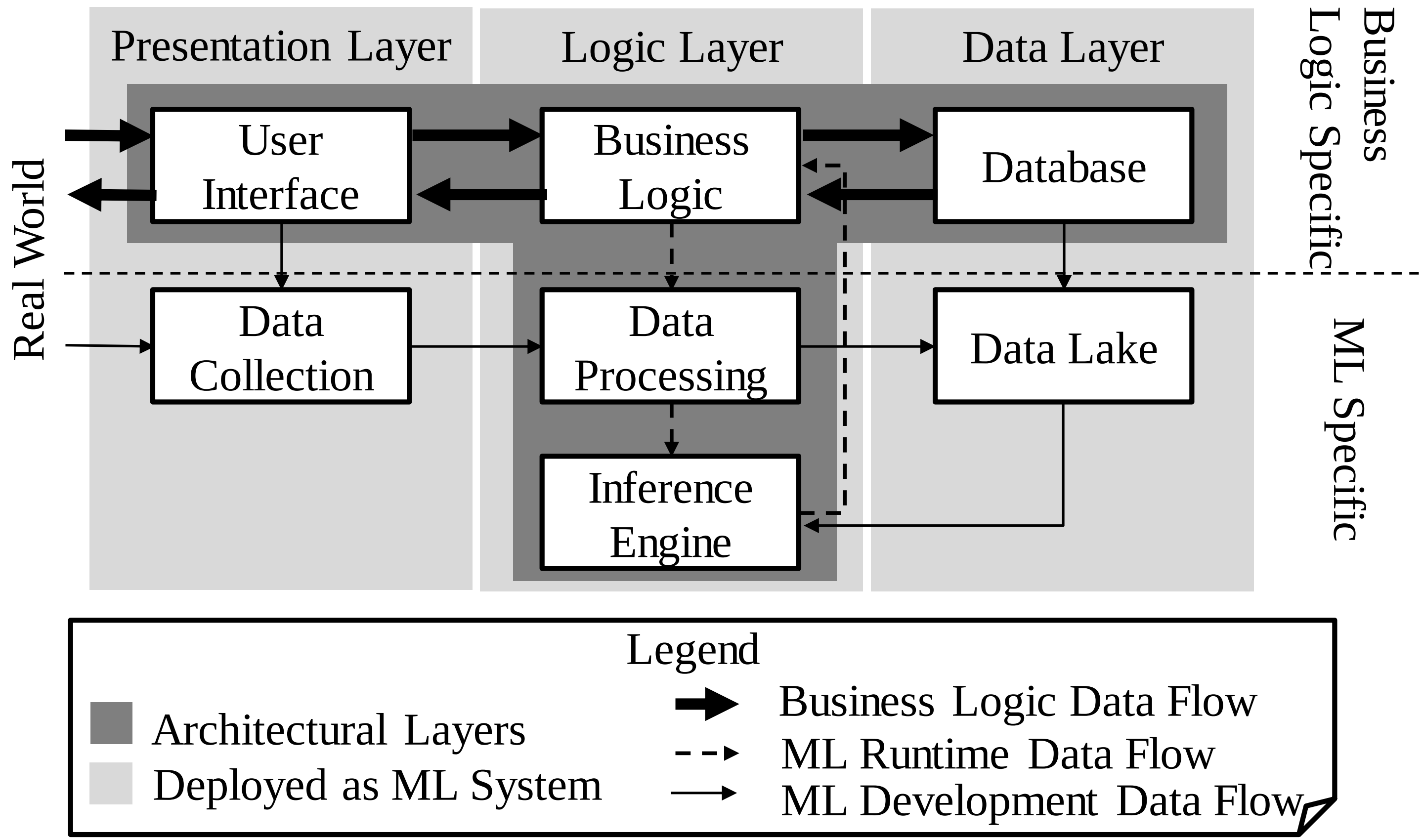
H. Washizaki, et al., Practitioners' insights on machine-learning software engineering design patterns: a preliminary study, ICSME 2020

Topology patterns

Pattern	Problem	Solution
Different Workloads in Different Computing Environments	It is necessary to separate and quickly change the ML data workload …	Physically isolate different workloads to separate machines…
Distinguish Business Logic from ML Models	The overall business logic should be isolated from the ML models …	Separate the business logic and the inference engine, loosely coupling the business logic and ML-specific dataflows.
ML Gateway Routing Architecture	Difficult to set up and manage individual endpoints for each service…	Install a gateway before a set of applications …
Microservice Architecture for ML	ML applications may be confined to some "known" ML frameworks …	Provide well-defined services to use for ML frameworks….
Lambda Architecture for ML	Real-time data processing requires scalability, fault tolerance, predictability …	The batch layer keeps producing views while the speed layer creates the relevant real-time views …
Kappa Architecture for ML	It is necessary to deal with huge amount of data with less code resource …	Support both real-time data processing and continuous reprocessing with a single stream processing engine …

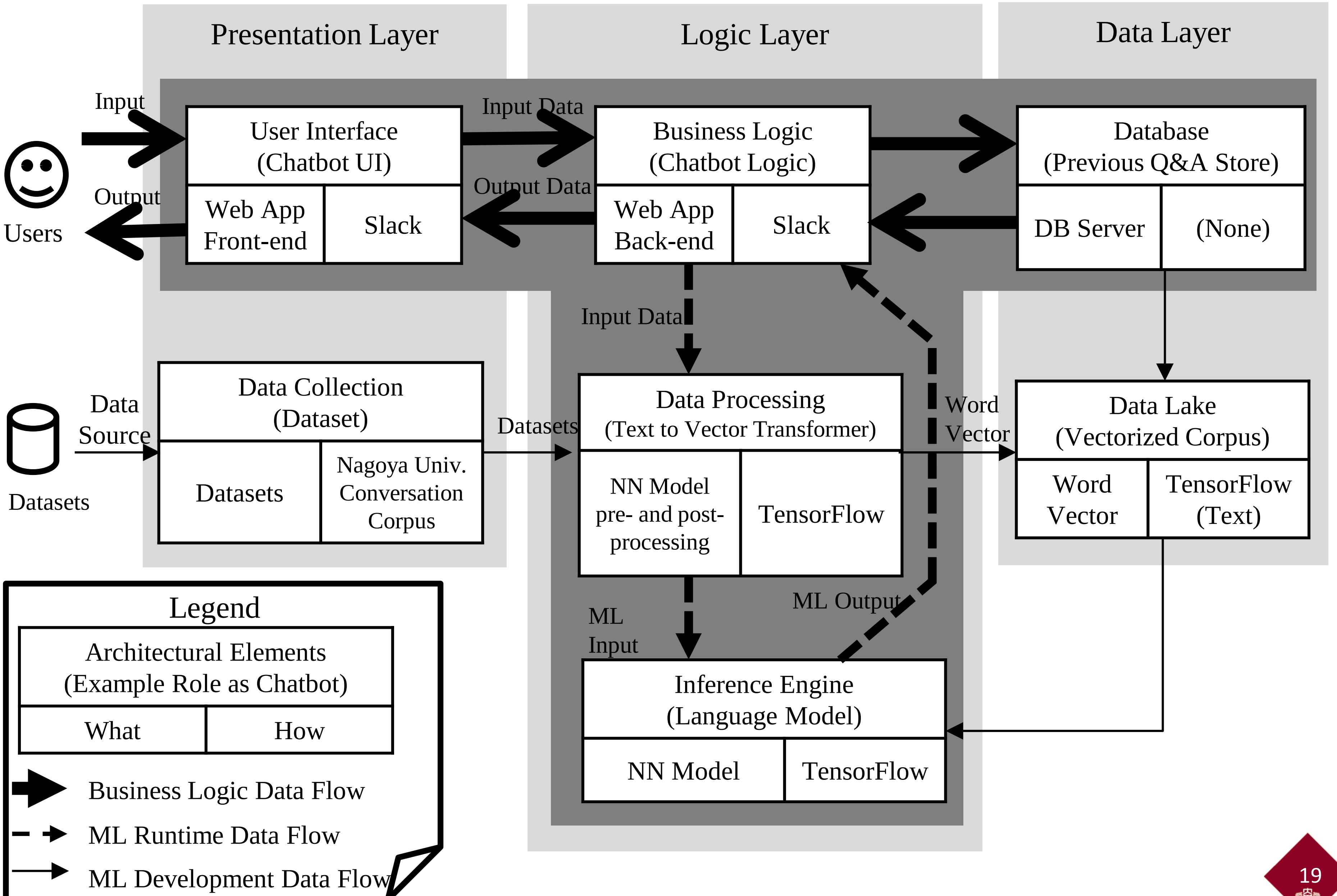
Distinguish Business Logic from ML Models

- **Problem:** Business logic should be isolated from ML models so that they can be changed without impacting rest of business logic.
- **Solution:** Separate the business logic and the inference engine, loosely coupling the business logic and ML-specific dataflows.



H. Yokoyama, Machine Learning System Architectural Pattern for Improving Operational Stability, ICSA-C, 2019
H. Washizaki, et al., Software Engineering Patterns for Machine Learning Applications (SEP4MLA), AsianPLoP 2020

Usage of **Distinguish Business Logic from ML Models**

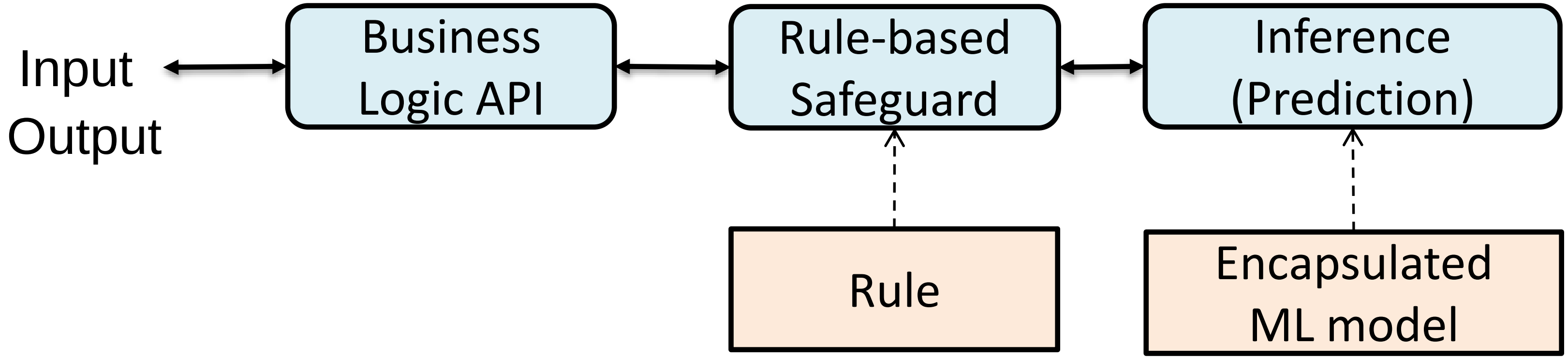


Programming patterns

Pattern	Problem	Solution
Data Lake for ML	We cannot foresee the kind of analyses that will be performed on the data …	Store data, which range from structured to unstructured, as “raw” as possible into a data storage …
Separation of Concerns and Modularization of ML Components	ML applications must accommodate regular and frequent changes to their ML components …	Decouple at different levels of complexity from the simplest to the most complex …
Encapsulate ML Models within Rule-based Safeguards	ML models are known to be unstable and vulnerable to adversarial attacks, drifts, …	Encapsulate functionality in the containing system using deterministic and verifiable rules …
Discard PoC Code	The code created for Proof of Concept (PoC) often includes code that sacrifices maintainability …	Discard the code created for the PoC and rebuild maintainable code …

Encapsulate ML Models within Rule-based Safeguards

- **Problem:** ML models are known to be unstable and vulnerable to adversarial attacks, noise, and data drift.
- **Solution:** Encapsulate functionality provided by ML models and deal with the inherent uncertainty in the containing system using deterministic and verifiable rules.
- Know usage: E.g. Apollos's object detection [Peng20]



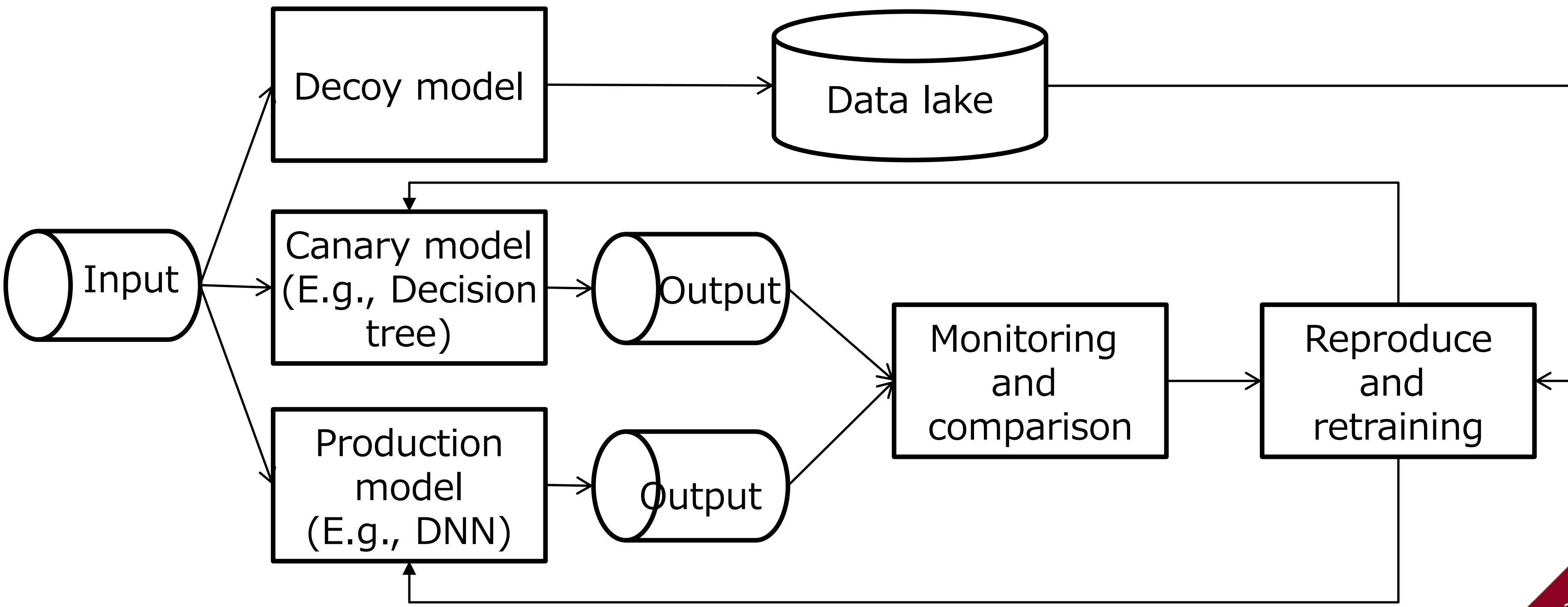
H. Washizaki, et al. Software Engineering Patterns for Machine Learning Applications (SEP4MLA) – Part 2, PLoP'20
Z. Peng, et al., A First Look at the Integration of Machine Learning Models in Complex Autonomous Driving Systems, ESEC/FSE'20

Model operation patterns

Pattern	Problem	Solution
Parameter-Server Abstraction	For distributed learning, widely accepted abstractions are lacking …	Distribute both data and workloads over worker nodes, while the server nodes maintain globally shared parameters …
Data Flows Up, Model Flows Down	Standard ML approaches require centralizing the training data on one machine …	Enable mobile devices to collaboratively learn while keeping all the training data on the device as federated learning …
Secure Aggregation	The system needs to communicate and aggregate model updates in a secure and scalable way …	Encrypt data from each device and calculate totals and averages without individual examination …
Deployable Canary Model	A surrogate ML that approximates the behavior of best model must be built to provide explainability …	Run the explainable inference pipeline in parallel to monitor prediction differences …
ML Versioning	ML models and their different versions may change the behavior of the overall ML applications …	Record the ML model, dataset, and code to ensure a reproducible training and inference processes …

Deployable Canary Model

- **Problem:** A surrogate ML that approximates the behavior of the best ML model must be built to provide explainability.
- **Solution:** Run the explainable inference pipeline in parallel with the primary inference pipeline to monitor prediction differences.
- Known usage: Image-based anomaly detection at factory



S. Ghanta et al., Interpretability and reproducibility in production machine learning applications, ICMLA 2018

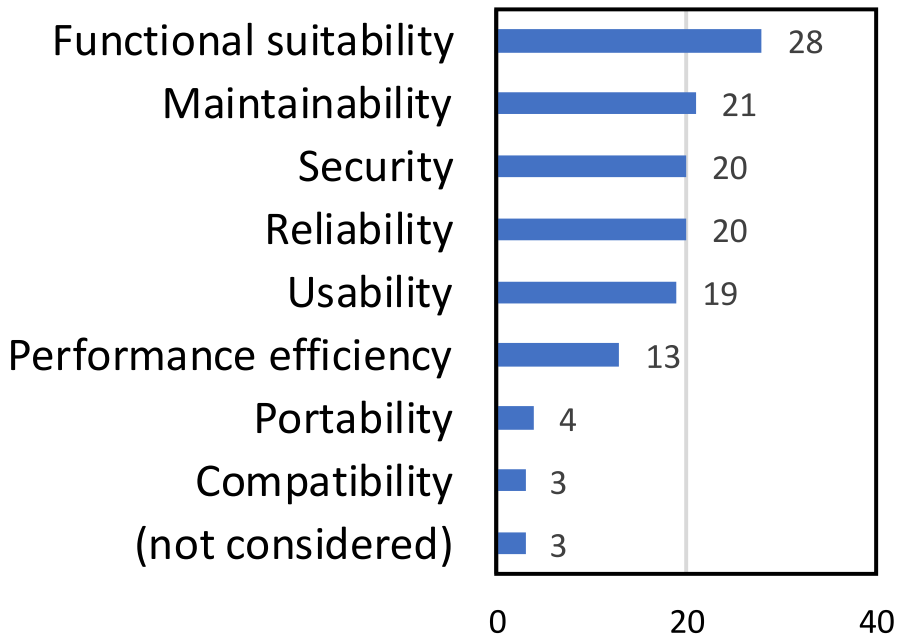
Pattern	Performance	Compatibility	Reliability	Security	Maintainability	Portability	Robustness	Explainability	Accuracy
Different Workloads in Different Computing Environments	X				X				
Distinguish Business Logic from ML Models					X				
ML Gateway Routing Architecture		X			X				
Microservice Architecture for ML		X			X	X			
Lambda Architecture for ML	X		X						
Kappa Architecture for ML	X		X						
Data Lake for ML	X	X			X				
Separation of Concerns and Modularization of ML Components					X				
Encapsulate ML Models within Rule-based Safeguards			X						
Discard PoC Code					X				
Parameter-Server Abstraction	X		X						
Data Flows Up, Model Flows Down	X						X		X
Secure Aggregation				X			X		X
Deployable Canary Model			X					X	
ML Versioning					X		X		X

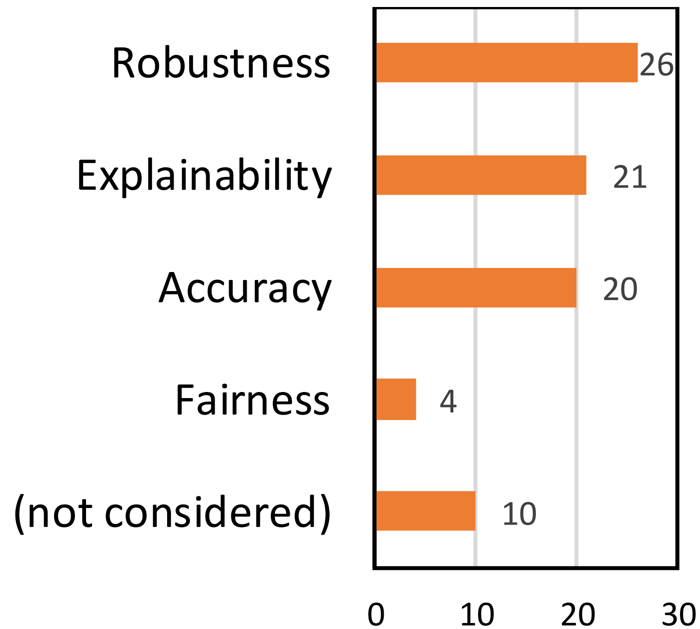
Agenda

- ML software engineering and patterns
- Literature review of software engineering patterns for ML applications
- Classification of ML patterns
- Practitioners' insights on ML patterns

RQ3. Practitioners' insights on quality

- Surveyed 300+ developers, 46 answered in ML development
- **What product quality attributes considered?**
  - Maintainability, reliability, security, and usability
- **What model and prediction quality attributes?**
  - Robustness, accuracy, and explainability
- Maintainability, reliability, robustness and accuracy are well handled by ML patterns. **There are demands for having ML patterns addressing security, usability, and explainability, which are not handled well now.**

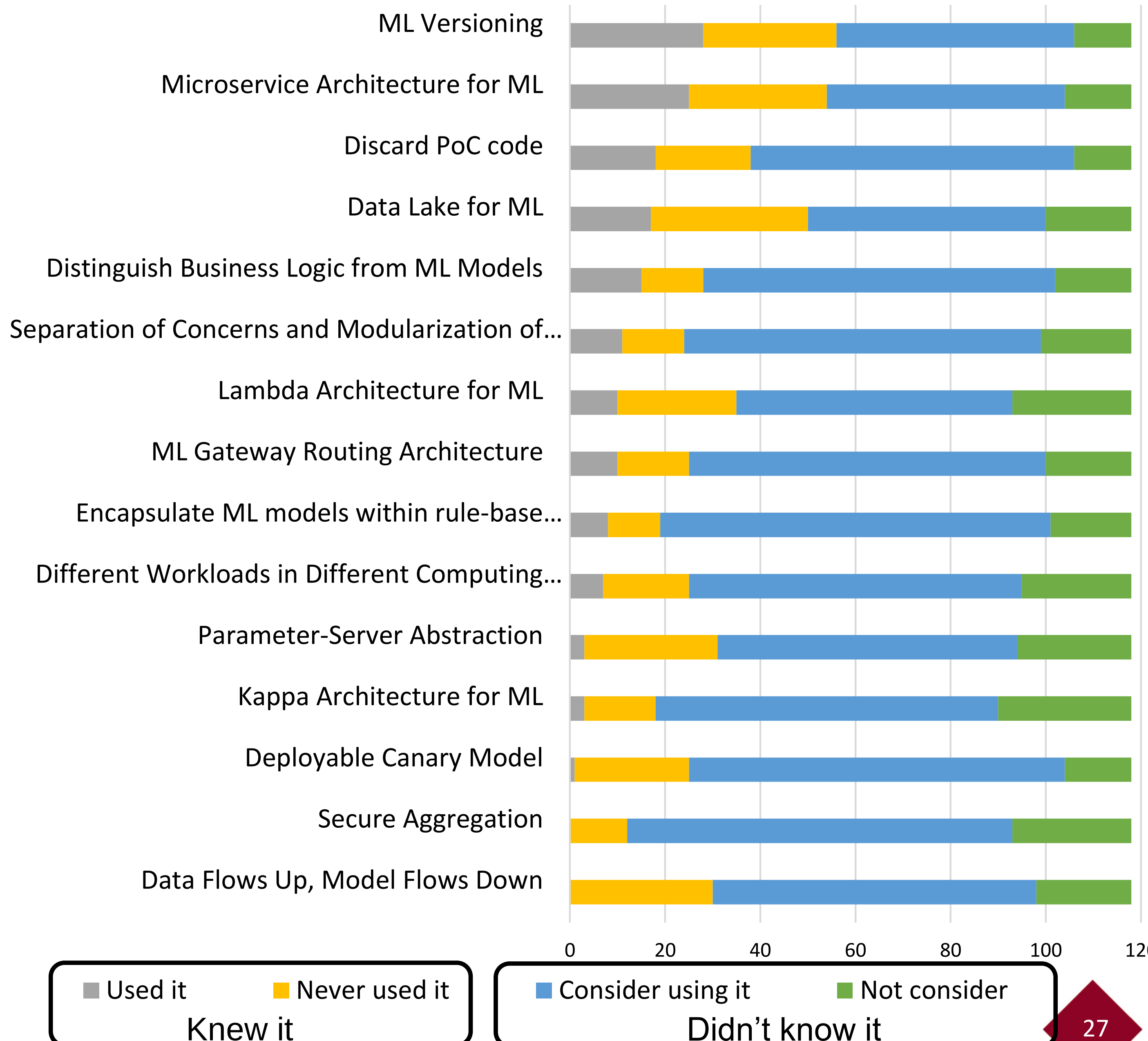




H. Washizaki, et al., Practitioners' insights on machine-learning software engineering design patterns: a preliminary study, ICSME 2020

Practitioners' insights on ML design patterns

- Surveyed 600+ developers, 118 answered
- **Have you ever referred to ML patterns?**
  - Major: **ML Versioning, Microservice Architecture for ML**
  - None: **Secure Aggregation, Data Flows Up** (aka. Federated Learning)



Practitioners' insights on ML design patterns

- **Have you ever referred to ML patterns?**
  - Developers were unfamiliar with most ML patterns, although there were several major patterns used by 20+% of the respondents.
  - For all patterns, most respondents indicated that they would consider using them in future designs.
  - Promoting existing ML patterns will increase their utilization
- **How do you solve and share design challenges of ML application systems?**
  - 37 (i.e., 31%) organized design patterns and past design results.
  - As respondents become more organized in their approach to design problems by reuse, the pattern usage ratio increased.
  - Development teams and organizations will reuse more ML patterns as they become more consistent in their reuse approach.

Design solution and reuse practice	#Respondents	#Patterns used	Pattern usage ratio
Lv3. Organizing, reusing patterns (and past results)	37	64	**11.5%**
Lv2. Reusing externally documented patterns	31	50	10.8%
Lv1. Resolving problems in an ad-hoc way	37	35	**6.3%**
Others	13	3	1.5%

Conclusion

- ML software engineering needs patterns!
  - As bride and common language
- Literature review of academic and gray literature
  - 19 scholarly and 19 gray documents identified.
  - 15 SE patterns for ML applications extracted.
  - Patterns at https://eai-transfer.github.io/ml-design-pattern/en/
  - ML patterns can be classified by scopes and quality attributes
- Survey of practitioners' insights
  - Developers were unfamiliar with most ML patterns, although there were several major patterns used by 20% (such as **ML Versioning** and **Microservice Architecture for ML)**
  - 31% organized design patterns and past design results.
  - As respondents become more organized in their approach to design problems by reuse, the pattern usage ratio increased.

Future research direction

- Identify ML patterns addressing specific quality attributes that are not handled well now
  - Security, usability, and explainability
- Investigate the impact of patterns on quality attributes of systems
- Analyze relationships among patterns including related ones towards a pattern language
  - V. Lakshmanan et al., “Machine Learning Design Patterns,” O’Reilly, 2020.
  - Y. Shibui, “Machine Learning System Design Patterns”, https://github.com/mercari/ml-system-design-pattern, 2020.
- Integration into framework to handle from requirements ton implementations and testing/debugging

Jati H. Husen, Hnin Thandar Tun, Nobukazu Yoshioka, Hironori Washizaki and Yoshiaki Fukazawa, “Goal-Oriented Machine Learning-Based Component Development Process,” ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS), Poster