Practitioners' insights on machine-learning software engineering design patterns: a preliminary study

Hironori Washizaki
Waseda University
washizaki@waseda.jp

Hironori Takeuchi
Musashi University

Foutse Khomh
Polytechnique Montréal

Naotake Natori
AISIN SEIKI Co.,Ltd.

Takuo Doi
Lifematics Inc.

Satoshi Okuda
Japan Advanced Institute of Science and Technology

***Abstract*—Machine-learning (ML) software engineering design patterns encapsulate reusable solutions to commonly occurring problems within the given contexts of ML systems and software design. These ML patterns should help develop and maintain ML systems and software from the design perspective. However, to the best of our knowledge, there is no study on the practitioners' insights on the use of ML patterns for design of their ML systems and software. Herein we report the preliminary results of a literature review and a questionnaire-based survey on ML system developers' state-of-practices with concrete ML patterns.**

***Index Terms*—Machine Learning, Design Patterns, Systematic Literature Review, Questionnaire Survey**

I. Introduction

Practitioners and researchers study best practices to design ML application systems and software to address quality and constraint problems. Such practices are often formalized as ML software engineering design patterns (hereafter, ML patterns), which encapsulate reusable solutions to address common problems within the given contexts of ML systems and software design. ML patterns should be useful for the development and maintenance of ML systems and software from the design perspective. However, a systematic study on how practitioners perceive and utilize ML patterns in the design of ML systems and software has yet to be conducted.

Herein we report the preliminary results of a literature review and a questionnaire survey on the practice and ML patterns. Section II describes our literature review and identified ML patterns. Section III describes our survey. Section IV summarizes related works. Section V concludes this paper.

II. ML software engineering design patterns

In [1], we performed a multivocal literature review of both formal literature and gray literature to collect software engineering patterns for ML application systems and software design. We extracted 33 ML design pattern candidates from 32 scholarly documents and 48 gray documents [1]. In this paper, we (including industrial ML developers) additionally reviewed the candidates from the viewpoint of practical usefulness, and identified 15 ML patterns. These patterns are grouped into three categories (Table I): ML systems topology patterns that define the entire system architecture, ML system programming patterns that define the design of particular component, and ML system model operation patterns that focus on ML models.

TABLE I
Identified ML Patterns

ID	Pattern name	Quality
Category: Topology		
P_1	Different Workloads in Different Computing Environments [4], [5]	E, M
P_2	Distinguish Business Logic from ML Models [6], [7]	M
P_3	ML Gateway Routing Architecture [6]	C, M
P_4	Microservice Architecture [7]–[9]	C, M, P
P_5	Lambda Architecture [10]–[12]	E, R
P_6	Kappa Architecture [13]	E, R
Category: Programming		
P_7	Data Lake [7], [10], [14], [15]	E, C, M
P_8	Separation of Concerns and Modularization of ML Components [16]	M
P_9	Encapsulate ML models within Rule-base Safeguards [5], [17]	R, S
P_{10}	Discard PoC Code [18]	M
Category: Model operation		
P_{11}	Parameter-Server Abstraction [18]	E, R
P_{12}	Data flows up, Model flow down [5], [19]	E, Mr, Pa
P_{13}	Secure Aggregation [19]	S, Mr, Pa
P_{14}	Deployable Canary Model [20]	R, Me
P_{15}	ML Versioning [4], [7], [18], [21]	M, Mr, Pa

Any design pattern is expected to address one or more quality attributes [2] that are associated with design problems. For ML patterns, we assumed that the following product quality attributes defined in ISO/IEC 25010:2011 [3] as well as model and prediction quality attributes can be addressed.

- System and software product quality attributes: Functional suitability (F), performance efficiency (E), compatibility (C), usability (U), reliability (R), security (S), maintainability (M), and, portability (P)
- ML model and prediction quality attributes: Model robustness (Mr), model explainability (Me), prediction accuracy (Pa), and, prediction fairness (Pf)

We analyzed 15 ML patterns by reviewing patterns' problem and solution descriptions. We identified that many ML patterns address maintainability. And most of model operation patterns address model and prediction quality attributes.

III. Survey

We surveyed 300+ software and ML developers who participated in an online seminar on ML patterns in July 2020. During the seminar, we explained the concept of software patterns and introduced the 15 ML patterns. Afterwards developers answered the following questions about reuse practices, quality attributes and patterns anonymously.

SQ1. How do you solve and share design challenges of machine learning (ML) application systems?

2576-3148/20/$31.00 ©2020 IEEE
DOI 10.1109/ICSME46990.2020.00095

SQ2. What product quality attributes do you consider when designing ML systems?

SQ3. What model and prediction quality attributes do you consider when designing ML systems?

SQ4. Have you ever referred to ML pattern P_i?

Out of the 300+ participants, 52 answered our questions, which corresponds to a response rate of around 17%. Table II summarizes the survey result of SQ1. Of the 46 participants involved in ML systems and software design, 20 (=2+10+8, 43%) reused past solutions in the form of internal-standards or patterns within or across teams.

Figure 1(left) summarizes the survey result of SQ2 for the 46 respondents. Most considered the functional suitability of the ML systems and software during design. This seems natural since functionality is the most fundamental attribute of any system and software. In addition, more than 40% of the respondents considered maintainability, reliability, security, and usability of the ML systems and software. In contrast, portability and compatibility were rarely considered. According to our pattern analysis in Table I, maintainability and reliability are well addressed in existing ML patterns while security and usability are less addressed; more ML patterns focusing on security and usability are anticipated by accumulating more design cases since these attributes are majorly concerned.

Figure 1(right) summarizes the result of SQ3. The top concern was model robustness, followed by model explainability and prediction accuracy. According to our pattern analysis, model robustness and prediction are well addressed in existing ML patterns but model explainability is less addressed.

Figure 2 summarizes the result of SQ4. The most used patterns were P_{15} (used by 22% of the respondents) followed by P_{10} and P_7 (20%). In terms of the use rate calculated by $\#Used/\#Knew$, P_2 is the most frequently used pattern with its use rate=8/13=0.62. On the other hand, P_7 was the most known but underused pattern. The popularity of P_{15}, P_{10} and P_2 is not surprising since they address an important quality attribute (i.e., maintainability) as shown in Figure 1. In contrast, no respondents actually used P_{12} or P_{13}. In summary, the developers were unfamiliar with most ML patterns, although there are several major patterns used by 20% of the respondents. For all patterns, most of the respondents indicated that they consider using them in future designs although they didn't know them. These findings suggest that there are opportunities to utilize existing ML patterns by increasing awareness of such patterns within the ML community.

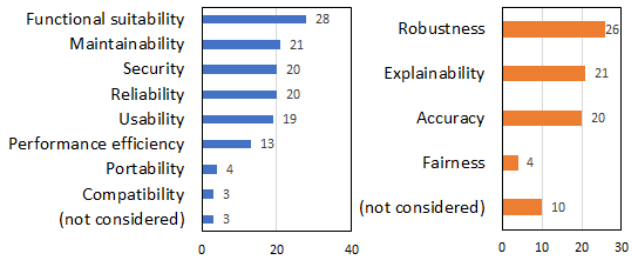

Fig. 1. Left: Result of SQ2, Right: Result of SQ3 (N=46, multiple choices)

TABLE II
SURVEY RESULT OF SQ1 (N=52)

Solution and reuse practice	#Answers
Solutions are standardized and reused across teams	2
Solutions are formulated as patterns and reused within the team	10
Patterns externally documented are reused	8
Results of previous designs are reused within the team	9
Design problems are resolved in an adhoc way without patterns	15
Other	2
NA (not involved in ML system design)	6

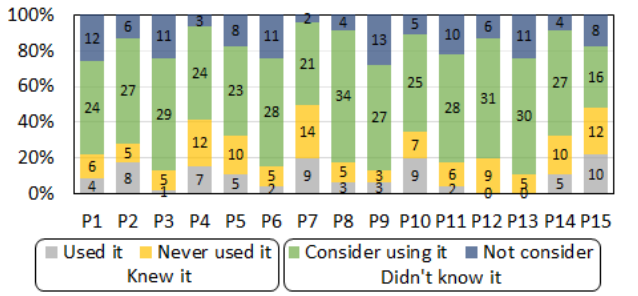

Fig. 2. Survey result of SQ4 (N=46): Knew and used it, Knew but never used it, Didn't know but consider using it in the future, and, Didn't know and not consider using it in the future

IV. RELATED WORK

There are surveys and case studies on practitioners' insights on ML systems development in general [16], [21]–[24]; however, none of them focus on the use of ML patterns.

After our initial literature review in 2019, several catalogs of ML design patterns have emerged [25], [26]. Although they have yet to be fully documented in the standard pattern format, practitioners may have explicitly or implicitly known and reused them. Thus, we will continue our survey by extending the scope of ML patterns including these catalogs.

V. CONCLUSION AND FUTURE WORK

Here, the preliminary results of our literature review and our questionnaire survey on ML developers' insights on the use of ML patterns in ML systems and software design are reported. We identified and analyzed 15 ML patterns through our literature review. 43% of the survey respondents have reused past solutions in the form of internal-standards or patterns. We confirmed that most of ML patterns identified by our literature review were unknown to the developers, although there are several major patterns ("Data Lake", "Discard PoC Code" and "ML Versioning") already used by around 20% of the respondents. Promoting existing ML patterns within the ML community will increase their utilization.

In the future, we will expand the list of ML patterns, conduct additional surveys and semi-structured interviews on past solutions and state-of-practices, and investigate applications of ML patterns in actual ML systems and software design.

ACKNOWLEDGEMENT

This work was supported by JSPS Bilateral Program JPJSBP120209936, JST-Mirai Program Grant Numbers JP18077318 and JP20319852, and enPiT-Pro Smart SE.

REFERENCES

[1] H. Washizaki, H. Uchida, F. Khomh, and Y. Guéhéneuc, "Studying software engineering patterns for designing machine learning systems," in *10th International Workshop on Empirical Software Engineering in Practice, IWESEP 2019, Tokyo, Japan, December 13-14, 2019*. IEEE, 2019, pp. 49–54. [Online]. Available: https://doi.org/10.1109/IWESEP49350.2019.00017

[2] H. Washizaki, S. Ogata, A. Hazeyama, T. Okubo, E. B. Fernandez, and N. Yoshioka, "Landscape of architecture and design patterns for iot systems," *IEEE Internet of Things Journal*, pp. 1–11, 2020.

[3] International Organization for Standardization, "ISO/IEC 25010:2011 systems and software engineering — systems and software quality requirements and evaluation (SQuaRE) – system and software quality models," 2011.

[4] C. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. M. Hazelwood, E. Isaac, Y. Jia, B. Jia, T. Leyvand, H. Lu, Y. Lu, L. Qiao, B. Reagen, J. Spisak, F. Sun, A. Tulloch, P. Vajda, X. Wang, Y. Wang, B. Wasti, Y. Wu, R. Xian, S. Yoo, and P. Zhang, "Machine learning at facebook: Understanding inference at the edge," in *25th International Symposium on High Performance Computer Architecture*. Washington, DC, USA: IEEE CS Press, 2019, pp. 331–344. [Online]. Available: https://doi.org/10.1109/HPCA.2019.00048

[5] H. Washizaki, F. Khomh, Y.-G. Guéhéneuc, H. Takeuchi, S. Okuda, N. Natori, Naohisa, and Shioura, "Software engineering patterns for machine learning applications (sep4mla) - part 2," in *(under review)*, 2020, pp. 1–10.

[6] H. Yokoyama, "Machine learning system architectural pattern for improving operational stability," in *International Conference on Software Architecture Companion*. Hamburg, Germany: IEEE CS Press, 2019, pp. 267–274. [Online]. Available: https://doi.org/10.1109/ICSA-C.2019.00055

[7] H. Washizaki, F. Khomh, and Y.-G. Guéhéneuc, "Software engineering patterns for machine learning applications (sep4mla)," in *9th Asian Conference on Pattern Languages of Programs (AsianPLoP 2020)*. Hillside, Inc., 2020, pp. 1–10.

[8] J. Everett, "Daisy architecture," https://datalanguage.com/features/daisy-architecture, July 2018.

[9] D. Smith, "Exploring development patterns in data science," https://www.theorylane.com/2017/10/20/some-development-patterns-in-data-science/, October 2017.

[10] P. Menon, "Demystifying data lake architecture," https://www.datasciencecentral.com/profiles/blogs/demystifying-data-lake-architecture, August 2017.

[11] "Lambda architecture pattern," https://hub.packtpub.com/lambdaarchitecture-pattern/.

[12] A. Basak, *Stream Analytics with Microsoft Azure: Real-time data processing for quick insights using Azure Stream Analytics*. Packt Publishing, 2017.

[13] "From insights to value - building a modern logical data lake to drive user adoption and business value," https://www.slideshare.net/Hadoop_Summit/from-insights-to-value-building-a-modern-logical-data-lake-to-drive-user-adoption-and-business-value.

[14] S. Gollapudi, *Practical Machine Learning*, ser. Community experience distilled. Birmingham, UK: Packt Publishing, 2016. [Online]. Available: https://books.google.ca/books?id=3ywhjwEACAAJ

[15] A. Singh, "Architecture of data lake," https://datascience.foundation/sciencewhitepaper/architecture-of-data-lake, April 2019.

[16] M. S. Rahman, E. Rivera, F. Khomh, Y. Guéhéneuc, and B. Lehnert, "Machine learning software engineering in practice: An industrial case study," *CoRR*, vol. abs/1906.07154, 2019. [Online]. Available: http://arxiv.org/abs/1906.07154

[17] M. Kläs and A. M. Vollmer, "Uncertainty in machine learning applications: A practice-driven classification of uncertainty," in *Computer Safety, Reliability, and Security - SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, Proceedings*, 2018, pp. 431–438. [Online]. Available: https://doi.org/10.1007/978-3-319-99229-7_36

[18] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in *Annual Conference on Neural Information Processing Systems*. Montréal, QC, Canada: Neural Information Processing Systems Foundation, 2015, pp. 2503–2511. [Online]. Available: http://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems

[19] Google, "Federated learning: Collaborative machine learning without centralized training data," https://ai.googleblog.com/2017/04/federated-learning-collaborative.html, 2017.

[20] "A design pattern for explainability and reproducibility in production ml," https://www.parallelm.com/a-design-pattern-for-explainability-and-reproducibility-in-production-ml/.

[21] S. Amershi, A. Begel, C. Bird, R. DeLine, H. C. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann, "Software engineering for machine learning: a case study," in *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE (SEIP) 2019, Montreal, QC, Canada, May 25-31, 2019*, 2019, pp. 291–300. [Online]. Available: https://doi.org/10.1109/ICSE-SEIP.2019.00042

[22] F. Ishikawa and N. Yoshioka, "How do engineers perceive difficulties in engineering of machine-learning systems?: questionnaire survey," in *Proceedings of the Joint 7th International Workshop on Conducting Empirical Studies in Industry and 6th International Workshop on Software Engineering Research and Industrial Practice, CESSER-IP@ICSE 2019, Montreal, QC, Canada, May 27, 2019*, M. Ciolkowski, D. Marijan, M. Galster, W. Shang, A. Jedlitschka, R. Shukla, and K. Padmanabhan, Eds. IEEE / ACM, 2019, pp. 2–9. [Online]. Available: https://dl.acm.org/citation.cfm?id=3338708

[23] L. E. Lwakatare, A. Raj, J. Bosch, H. H. Olsson, and I. Crnkovic, "A taxonomy of software engineering challenges for machine learning systems: An empirical investigation," in *Agile Processes in Software Engineering and Extreme Programming - 20th International Conference, XP 2019, Montréal, QC, Canada, May 21-25, 2019, Proceedings*, ser. Lecture Notes in Business Information Processing, P. Kruchten, S. Fraser, and F. Coallier, Eds., vol. 355. Springer, 2019, pp. 227–243. [Online]. Available: https://doi.org/10.1007/978-3-030-19034-7_14

[24] H. Kuwajima, H. Yasuoka, and T. Nakae, "Engineering problems in machine learning systems," *Mach. Learn.*, vol. 109, no. 5, pp. 1103–1126, 2020. [Online]. Available: https://doi.org/10.1007/s10994-020-05872-w

[25] V. Lakshmanan, S. Robinson, and M. Munn, "Machine learning design patterns," https://www.oreilly.com/library/view/machine-learning-design/9781098115777/, December 2020.

[26] Y. Shibui, "Machine learning system design patterns," https://github.com/mercari/ml-system-design-pattern, June 2020.