Preliminary Systematic Literature Review of Machine Learning System Development Process

Yasuhiko Watanabe¹, Hironori Washizaki², Kazunori Sakamoto³, Daisuke Saito⁴, Kiyoshi Honda⁵, Naohiko Tsuda⁶, Yoshiaki Fukazawa⁷ and Nobukazu Yoshioka⁸

¹²³⁴⁵⁶⁷ Waseda University, Tokyo, Japan
⁵ Osaka Institute of Technology, Osaka, Japan
⁸ National Institute of Informatics, Tokyo, Japan
{¹jellyfish44-time@akane., ²washizaki@, ³k.sakamoto@ai.oni., ⁴d.saito@fuji., ⁶821821@toki., ⁷fukazawa@} waseda.jp,
⁵kiyoshi.honda@oit.ac.jp, ⁸nobukazu@nii.ac.jp

Abstract—Previous machine learning (ML) system development research suggests that emerging software quality attributes are a concern due to the probabilistic behavior of ML systems. Assuming that detailed development processes depend on individual developers and are not discussed in detail. To help developers to standardize their ML system development processes, we conduct a preliminary systematic literature review on ML system development processes. A search query of 2358 papers identified 7 papers as well as two other papers determined in an ad-hoc review. Our findings include emphasized phases in ML system developments, frequently described practices and tailored traditional software development practices.

Index Terms—machine learning, software engineering, process, systematic literature survey

I. INTRODUCTION

Machine learning (ML) systems are complex systems. Machine learning algorithm’s behaviors are probabilistic because they depend on training data. In contrast, the behaviors in traditional software development are defined by code.

Some researchers suspect that the probabilistic behaviors derive emerging quality or validation concerns [¹]–[³]. Additionally, other studies have investigated ML system’s bugs derived from algorithms, data dependency and its architectures [⁴, ⁵]. Moreover, data infrastructure is also complex because it often deals with big data [⁶]. Quality aspects of ML systems are well documented. A literature review is conducted [⁷]. ML system development challenges are also discussed by conducting empirical case study [⁸]. However, development practices are not discussed well. We assume that these practices depend on individual developers or organizations. It prevents developers from utilizing practices to ML system developments.

To understand current practices and to help developers standardize ML system development processes, we implement a preliminary systematic literature review. We defined a search query in Scopus for the review. The query returned 2358 papers as potential hits. Of those, 7 papers are related to ML system development process. We reviewed and compared these seven papers to two other papers which were retrieved by an ad-hoc review. This paper reveals seven main findings:

- F1: Model Training & Evaluation phases are frequently described in several papers,
- F2: Some phases are mentioned in specific communities,
- F3: Some issues and practices are described frequently,
- F4: ML system development employs practices in Data Concerns, Start Small or Measure category in several phases,
- F5: Practices in Separation of Concerns category are employed even in ML system development,
- F6: Practices in Goal-oriented category are tailored to Model Evaluation and
- F7: Traditional practices are useful even in ML system developments.

II. REVIEW METHOD

A. Research Questions

To help developers standardize ML system development processes, We defined the three research questions (RQs).

To understand how existing practices tackle issues in ML system developments, we investigated: RQ1: Which phases are emphasized in ML system developments?

To understand detailed activities, we investigated: RQ2: What kind of practices are included in ML system developments?

To help developers to utilize their current experiences even in ML system developments, we investigated: RQ3: Do practices in ML system developments include traditional software development practices?

B. Inclusion and Exclusion Criteria

We surveyed following articles which satisfy all following inclusion criteria:

- Indexed in Scopus,
- Related to Software engineering process for Machine Learning system,
- Related to processes, practices or role of developers in ML system development,
- Included in Engineering, Computer Science or Mathematics domain, and
- Published in January 2010 to March 2019

We excluded articles which satisfy any following exclusion criteria:

- Proposals of machine learning algorithm
Machine learning applications for software engineering tasks or a specific project
- Proposals of software engineering techniques without tailoring a process or phases (e.g., proposal of tool to improve efficiency of a specific phase)
- Quality concerns of ML systems

C. Search Strategies

We constructed the following query in Scopus to satisfy the above criteria. Our query in Scopus was ("machine learning" OR "machine-learning" OR "ML") AND ("software" OR "system" OR "systems") AND ("engineering") AND ("practice" OR "organization pattern" OR "process" OR "process pattern" OR "empirical" OR "empirical study" OR "case study" OR "field study") Its search targeted the title and abstract.

The query returned 2358 papers from Scopus. The initial results included papers related to machine learning algorithms, applications to a specific domain, machine learning utilization to software engineering, etc. A review by first author identified seven papers related to ML system development processes. Additionally, the first author found two related papers, which are not indexed in Scopus via a ad-hoc review. Thus the first author analyzed the nine papers and identified issues and practices contained within.

III. RESULTS AND FINDINGS

A. Overall findings

We surveyed nine papers (9–14 via Scopus and 15–17 via ad-hoc review). The surveyed papers include five papers based on interview with developers (9–11, 13, 16) and three papers proposed their own processes (12, 14, 15). Other one paper provides the best practices from industry (17). High distribution of interview papers implies that current research tried to figure out existing development processes in bottom-up by conducting interviews.

Five papers are published in computer science communities (9, 10, 13, 16, 17) and four papers are published in human-centered design communities (11, 12, 14, 15).

Reviewed studies assume that ML system development teams include the following roles: developers (9, 15, 17), data scientists (10, 13), UX designers (11), domain experts (14) and end-users (12, 15).

To answer RQ1, RQ2 and RQ3, We mapped issues and practices described in each papers into phases of Table I. Phases and Definitions in Table I show phases and their definitions in ML system development processes. Issues and Practices in Table I show issues and practices which we found in each phase by reviewing nine papers.

To answer RQ2 and RQ3, we defined categories of practices, their definitions and distribution of categories by phase (Table II). We found that several practices have the same purpose or characteristics even though these practices are employed in different phases. We defined categories based on some common purposes or characteristics in practices (Categories and Definitions in Table II) and categorized practices into a category by phase (Practices by Phase in Table II).

B. RQ1: Phases in ML system developments

As answer to RQ1, we explain two findings F1 and F2.

F1: Model Training & Evaluation phases are frequently described in several papers. Issues in Model Training (MT) phase are described in four papers (9, 10, 16, 17) (Table I). Issues in Model Evaluation (ME) phase are described in six papers (9–11, 13, 16, 17) (Table I). Practices in MT and ME phases are also described in several papers (Table I). Practices in MT are described in three papers (9, 13, 17) and practices in ME are described in four papers (9, 10, 12, 17) (Table I).

F2: Some phases are mentioned in specific communities. Issues in Model Requirement (MR) phase are mentioned by studies in human-centered design communities. Five issues in MR phase are described in two papers (11, 15) which are published only in human-centered design communities (Table I).

On the other hand, several data and model related phases are described only in computer science communities. Issues in Data Cleaning (DCl) & Data Labeling (Dl), practices in Data Collection (DCo), and issues and practices in Feature Engineering (FE), MT, & Model Deployment (MD) are mentioned by papers which are published only in computer science communities (Table I).

C. RQ2: Practices in ML system developments

As answer to RQ2, we explain two findings F3 and F4.

F3: Some issues and practices are described frequently. Several papers mentioned same issues or practices (Table I). Issue I71 in MT phase is described in three papers. Issues I22 & I23 in DCo phase, I72 in MT phase and I110 & I112 in Cross-Cutting (CC) phase are described in two papers. These frequent issues are included in DCo, MT or CC phase. Issues I110 and I112 are related to human capabilities. Practice P61 in MT phase is described in three papers. Practices P11, P73, P81, P91 and P101 are described in two papers. These frequent practices are included in MR, MT, ME, MD, or Model Monitoring (MM) and CC phases.

F4: Practices in Data Concerns, Start Small or Measure category are employed in many phases. Practices in Data Concerns category (Table I) are included in five phases (MR, DCo, FE, MT and MM). Additionally, there is no practice for DCo and DL phases. We expect that difficulties in data management are tackled but data collection and data labeling are still challenging.

Practices in Start Small or Measure category are also employed in many phases. Practices in Start Small or Measure category are included in four phases (Table III).

D. RQ3: Traditional software development practices

As answer to RQ3, we explain following three findings: F5, F6 and F7.

F5: Practices in Separation of concerns category are employed even in ML system development. Separation
Thus, we argue that practices in Goal-oriented category are tailored to Model Evaluation.

**F7: Traditional practices are useful even in ML system developments.** We found other five traditional software engineering practices described in reviewed papers (Traditional Practice in Table I). These five practices include requirement elicitation from users (P13 and P14), traceability management (P22), refactoring (P23), and testing (P82). Practices in Traditional Practice category are employed in MR, DCo and MD phases (Table II).

Practices in Separation of concerns, Goal-oriented or Traditional Practice category are employed in five (MR, DCo, MT, ME and MD) out of eight phases (Table II). Thus, we argue that traditional concepts or practices may be useful in even though issues in ML system have unique aspects.
### TABLE II
**Categories of practices**

| Practice Category | Category Definition | Practices by Phase (each phase is represented by its abbreviation) |
|-------------------|---------------------|---------------------------------------------------------------|
| Start Small       | Practices to start with simplified issues. | P21 |
| Traditional Practice | Practices employed even in traditional software developments | P13, P14, P22, P23 |
| Goal-oriented Data Concerns | Practices to focus on a goal of the project | P11 |
| Measure | Practices to measure uncertainty in developments | P12 |
| Heuristic | Practices which rely on developers’ experiences | P15 |

| Category | Practices | Phase (each phase is represented by its abbreviation) |
|----------|-----------|-----------------------------------------------------|
| DC | DCi | P53 P61 P83 |
| MR | DL | P81 |
| MT | ME | P91 |
| MD | MM | P74 |
| P62 | P72 P81 |
| P92 |

### IV. Threats to Validity

The inclusion and exclusion criteria may be a threat to validity. In this research, we reviewed papers which purely focused on ML system development processes. However, case study papers may possibly include discussion about development processes. Additionally, the boundary between data mining projects and ML system developments is vague, which may affect the authors’ paper selection.

### V. Conclusion

To understand current practices and to help developers standardize ML system development processes, we conducted a preliminary systematic literature review on ML system development processes.

This study reveals seven following findings: F1: Model Training & Evaluation phases are frequently described in several papers, F2: Some phases are mentioned in specific communities, F3: Some issues and practices are described frequently, F4: ML system development employs practices in Data Concerns, Start Small or Measure category in several phases, F5: Practices in Separation of Concerns category are employed even in ML system development, F6: Practices in Goal-oriented category are tailored to Model Evaluation and F7: Traditional practices are useful even in ML system developments.

In the future, we will conduct interviews with developers and discuss the effectiveness of each practice.

### VI. Acknowledgements

This work was supported by JST-Mirai Program Grant Number JP18077318, Japan.

### References

[1] A. Arpit et al., “Software engineering challenges of deep learning,” in SEAA 2018, Aug 2018, pp. 50–59.

[2] F. Khomh et al., “Software engineering for machine-learning applications: The road ahead,” IEEE Software, vol. 35, no. 5, pp. 81–84, Sep. 2018.

[3] Z. C. Lipton, “The mythos of model interpretability,” Queue, vol. 16, no. 3, pp. 30:31–30:57, Jun. 2018.

[4] D. Sculley et al., “Hidden technical debt in machine learning systems,” in NIPS 2015. Curran Associates, Inc., 2015, pp. 2503–2511.

[5] F. Thung et al., “An empirical study of bugs in machine learning systems,” in ISSRE 2012, Nov 2012, pp. 271–280.

[6] H. ZHAO and J. HOU, “Design concerns for industrial big data system in the smart factory domain:from product lifecycle view,” in ICECCS 2018, Aug 2018, pp. 50–59.

[7] S. Masuda et al., “A survey of software quality for machine learning applications,” in ICSTW 2018, April 2018, pp. 279–284.

[8] L. E. Lwakatare et al, “A taxonomy of software engineering challenges for machine learning systems: An empirical investigation,” in XP. Cham: Springer International Publishing, 2019, pp. 227–243.

[9] C. Hill et al., “Trials and tribulations of developers of intelligent systems: A field study,” in VL/HCC 2016, Sep. 2016, pp. 162–170.

[10] M. Kim et al., “The emerging role of data scientists on software development teams,” in Proceedings of ICSE, ser. ICSE ‘16. New York, NY, USA: ACM, 2016, pp. 96–107.

[11] G. Dove et al., “Ux design innovation: Challenges for working with machine learning as a design material,” in Proceedings of CHI 2017, ser. CHI ’17. New York, NY, USA: ACM, 2017, pp. 278–288.

[12] R. Fiebrink et al., “Human model evaluation in interactive supervised learning,” in Proceedings of CHI 2011, ser. CHI ‘11. New York, NY, USA: ACM, 2011, pp. 147–156.

[13] C. T. Wolf, Professional Identity and Information Use: On Becoming a Machine Learning Developer: 14th International Conference, iConference 2019, Washington, DC, USA, March 31April 3, 2019, Proceedings, 03 2019, pp. 625–636.

[14] T. Seymoens et al., “A methodology to involve domain experts and machine learning techniques in the design of human-centered algorithms,” in Human Work Interaction Design. Designing Engaging Automation. Cham: Springer International Publishing, 2019, pp. 200–214.

[15] A. Billewicz, “Silly lamp: Study of a relationship with engaging machine learning artefacts,” in Extended Abstracts of CHI 2018, ser. CHI EA ’18. New York, NY, USA: ACM, 2018, pp. SRC03:1–SRC03:6.

[16] S. Amershi et al., “Software engineering for machine learning: A case study,” in Proceedings of ICSE, ser. ICSE-SEIP ’19, 2019. [Online]. Available: https://www.microsoft.com/en-us/research/publication/software-engineering-for-machine-learning-a-case-study/

[17] M. Zinkevich, “Rules of machine learning: Best practices for ml engineering,”[http://martin.zinkevich.org/rules_of_ml/rules_of_ml.pdf](http://martin.zinkevich.org/rules_of_ml/rules_of_ml.pdf) 2017.

[18] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin, “Aspect-oriented programming,” in European conference on object-oriented programming. Springer, 1997, pp. 220–242.

[19] V. Basili et al., Aligning Organizations Through Measurement The GQM+Strategies Approach. Springer, 2014.