How Software Engineering Relates to Machine Learning

Sourasekhar Banerjee

1 Introduction

Federated learning (FL) is a specialized version of distributed machine learning that trains machine learning models collaboratively while keeping clients' data private. Clients are usually resource-constrained heterogeneous edge devices. It follows a client-server architecture where multiple clients are responsible for local computations. The cloud acts as a central parameter server where the client sends local models for aggregation and generates the global model. Federated learning has four stages, such as, (1) All clients install the client-side version of the software through which clients will send the learned parameters to the server. (2) The server sends the initial model to all the connected clients. (3) Clients learns the model and send it back to the server for aggregation. (4) After aggregation, the server sends the model to the clients for evaluation. These four steps continue until the model reaches the desired accuracy. In my Ph.D. project, we are exploring different characteristics and challenges (system and statistical heterogeneity) of federated learning and providing unique solutions to mitigate those challenges. The focused area of the research is on straggler mitigation and model personalization.

2 Topics

Here we will describe automated software testing, requirements engineering (RE), and project management. We will also discuss application of machine learning in software engineering.

2.1 Automated Software Testing

Automated testing is a software tool that automates the manual process of validating a software product [1]. It is a way to improve the effectiveness, execution speed, and test coverage of software testing. Automation's objective is to minimize the number of test cases that must be run manually, not completely replace manual testing. Repeated execution of the same test suite will be necessary during subsequent development cycles. This test suite can be recorded and replayed as needed using a test automation tool. The automation testing software may also generate thorough test reports and compare expected and actual findings. Automation in software testing is essential because (1) manual testing is time- and cost-effective and error-prone; (2) testing through automation doesn't require human intervention during software testing; and (3) automation increases test coverage and speed of test execution. Automated testing promotes lean quality assurance(QA) team size and enables the QA team to focus on more sensitive features. Tests, such as end-to-end, unit, integration, and performance tests, should be automated first. Automated testing is unsuitable in scenarios where requirements are frequently changing, or test cases are executed on an ad-hoc basis. For those cases, manual testing is required.

Machine learning (ML) based technologies enhance the automated software testing procedure [2]. Applications of ML, such as object detection, anomaly detection, pattern recognition, etc., can be extended to automated software design. For example, object detection can be applied in user interface testing automation where the human eye might still overlook certain faulty parts on the page. For API monitoring, anomaly detection [3] could help to detect unusual API events and traffic. Some tools that are useful in automated testing are DiffBlue, Facebook Infer, Google OSS-Fuzz, Launchable, etc.

2.2 Requirements Engineering

The primary goal of building a software system is to fulfill the users' expectations. One of the major components to accomplishing this goal is requirements engineering. It is the process of identifying software requirements in a systematic manner to understand what functionalities the software system should have to fulfill the users' needs. Requirements engineering has four phases: (1) requirements elicitation, (2) requirements analysis, (3) requirements documentation, and (4) requirement verification.

Errors in RE can be expensive in terms of lost time, money, reputation, and potentially the project's viability. Machine learning can improve the software development process and make it more efficient [4]. ML approaches automate the requirement engineering activities such as ambiguity detection, traceability analysis, etc., and try to reduce the cost and time of RE. The challenges that are faced by ML approaches in RE can be divided into five categories, such as: data-related, task-related, algorithm-related, project-related, and language-related. Data-related challenges are mostly in quantity, quality, and skewness that cause overfitting or underfitting of the model. The task-related challenges are mostly centered on classification and regression problems. Algorithmic challenges are related to the interpretability of the ML algorithm. Project-specific challenges impose limitations that prevent approaches from being generalizable. In language-related problems, inconsistency in requirements documentation makes automated classification more complicated and error-prone. Classical machine learning (KNN, Decision Tree, Random Forest, etc.) and deep neural networks (CNN, LSTM, Transformers, etc.) both are very popular nowadays in automating requirements engineering.

2.3 Project Management

Software project management is a method of leading software projects where projects are planned, implemented, controlled, and monitored. Basic technology changes and advances so frequently and rapidly that increases risk in software development. Hence, it is essential to manage software projects efficiently. Project management consists of conflict management, risk management, requirement management, change management, software configuration management, and release management.

There is a lot of potential for ML in project management. For example, ML could be used to help identify risks and issues early on in a project [5] or to help predict how likely it is that a project will be completed on time and within budget. Additionally, ML could be used to help optimize project schedules and resources, or to automatically generate project reports. ML based models are heavily used in industries to make better decisions and personalized actions [6]. Federated learning (FL) can be used in project management to train a machine learning model on data distributed across several projects. Project data can be stored on individual devices or multiple servers. The data on each device is used to train a local model. The local models are then combined to create a global model. For example, the global model prepared using FL could be used to predict the completion time of a project.

3 Future Trends and Directions of Software Engineering

In 2011, Marc Andreessen famously said, "Software is eating the world [7]." The companies that operated mainly in the real physical world were transitioning to the digital economy, transforming every company into a software company. Similarly, the influence of machine learning can be seen in software industries nowadays. Machine learning is being used in various industries, from healthcare to finance to manufacturing. Most software companies use ML-based solutions to increase efficiency and make better decisions. As machine learning technology becomes more sophisticated, it's beginning to eat into the software market. There are a few reasons for this. First, machine learning can be used to develop predictive models by learning from historical data that are far more accurate than traditional software models. Second, ML can be used to automate tasks that are traditionally done by software. For example, ML can be used to automatically identify patterns in data sets, which can then be used to make recommendations. Lastly, ML is becoming more accessible to businesses of all sizes. Thanks to cloud-based machine learning platforms, businesses no longer need to invest in expensive hardware and software to get started with ML. These factors lead to a future where machine learning will eat into the software market.

Data science is focused on the data that is being processed. The code is secondary. This is a significant difference from software development, where the focus is on the code that is written. Many companies are using data science and ML techniques as part of their software engineering processes. For example, Facebook uses ML to tag friends in photos automatically, Google to improve search results, and Netflix to recommend movies.

ML-based software must manage three main assets: data, models, and code. ML-Ops (Machine Learning Model Operationalization Management) is an extension of DevOps to establish the design, building, and deployment of ML models into production. The deployment architecture for trained ML models is a web service (microservices). The web service provides real-time predictions based on the provided input data points. The model remains constant until it is re-trained and re-deployed into the production system. Another exciting concept is called "online learning," where, in addition to making predictions from new data, a model can also learn from it. ML-based software systems follow a similar software development life cycle. In [8], the authors have shown the Federated learning (FL) system also follows a similar software engineering principle to make FL-based software products. In conclusion, data science and ML are going to be an integral part of software engineering in near future.

References

[1] "https://www.atlassian.com/continuous-delivery/software-testing/automated-testing," Last accessed Oct 13,2022.

[2] "https://www.browserstack.com/," Last accessed Oct 13,2022.

[3] "https://www.perfecto.io/blog/classifying-ai-testing-tools," Last accessed Oct 13,2022.

[4] K. Zamani, D. Zowghi, and C. Arora, "Machine learning in requirements engineering: A mapping study," in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pp. 116–125, IEEE, 2021.

[5] M. N. Mahdi, M. H. Mohamed Zabil, A. R. Ahmad, R. Ismail, Y. Yusoff, L. K. Cheng, M. S. B. M. Azmi, H. Natiq, and H. Happala Naidu, "Software project management using machine learning technique—a review," *Applied Sciences*, vol. 11, no. 11, p. 5183, 2021.

[6] "https://www.forecast.app/blog/machine-learning-in-project-management," Last accessed Oct 13,2022.

[7] "https://www.wsj.com/articles/sb10001424053111903480904576512250915629460," Last accessed Oct 13,2022.

[8] S. K. Lo, Q. Lu, C. Wang, H.-Y. Paik, and L. Zhu, "A systematic literature review on federated machine learning: From a software engineering perspective," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–39, 2021.