The extra mile to take: Technical Debt solutions for Industrial ML

Hadil Abukwaik[1], Benjamin Kloepper[1], Reuben Borrison[1] and Shrikant Bhat[2]

[1]ABB Corporate Research Center, Ladenburg, Germany
[2]ABB Corporate Research Center, Bangalore, India

While major studies on technical debt (TD) are on architecture and code of conventional software systems, very few focus on TD in ML like Sculley et al.[1,2]. Also, there is a particular need to tackle *TD in industrial ML* along its life cycle considering both the special nature of ML systems (e.g., it is data-driven, non-deterministic, and heavy-computation experimentation) and the restrictions inherited in industrial context (e.g., domain constraints, rare incidents, on-premise operation, and customer ownership of data). In an exploratory study conducted by four researchers with backgrounds in software engineering, software architecture, data science, ML engineering, and process automation. We interviewed 29 industrial experts who develop ML solutions for process automation in industrial domains like chemical, oil refineries, paper and pulp factories. We revealed various TD issues along the ML life cycle and reviewed existing practical solutions to proactively minimize the ML-specific TD. Here, we give pointers to some TD architectural solutions that fall short to be applicable in process automation industry.

Data architectural solutions facing rare data and contextual heterogeneity: Industrial applications have increasingly large amount of devices and collected data by sensors and controllers. Still, they suffer a severe class imbalance for incidents representing expensive and undesirable situations (e.g., plant shutdowns, pump trips, and safety interlocks). Also, industrial processes are contextually heterogeneous making it a real challenge for data reuse and sharing solutions (e.g., data mesh, feature stores, and API) adopted by big tech companies fueled by ML (e.g., Netflix recommendation and Google search).

Continuous integration facing Operational Technology (OT) restrictions: While mainstream momentum shows interest in achieving speed and agility in ML engineering (e.g., adopting MLOps), it is a hurdle to fit the continuous model integration in the existing OT stack (e.g., distributed control system, remote terminal units, and their HMI), the conservative release process, and the need for domain experts' validation for the solution.

Release patterns facing sufficient feedback: Using reliable patterns, like canary releases or A/B testing, in continuous delivery for ML (CD4ML) has an obstacle of rare feedback on deployed industrial ML. Anomalies come rarely, take too long to happen, spread across various operational systems, and get delayed labels (e.g., predicted pump failure comes available only after disassembly, transport to operation workshop and inspection). Also, in many industrial use cases ML models are not generalizable and the patterns' advantage of eliminating downtime or disruptions for many end-users are limited.

Data management facing data access: Adopting data management platforms and DataOps has a stretched challenge in industry represented in data ownership and the need for trust-serving design and protocols to share data between producers (e.g., chemical plant) and consumers (e.g., ML-based automation solution provider).

Adaptive ML facing industrial constraints: Quality degradation in industrial ML due to data drift has significant impact on environment and production. This imposes the need for conservative design of adaptation criteria and thresholds. For example, adapting data collection frequency should consider not only the ML application concerns (e.g., responses time or precision), but also industrial constraints (e.g., time to change tank temperature) and infrastructure restrictions (e.g., IIoT computational power). This requires a maturity level in continuous monitoring and feedback for ML in production, which does not apply to many industrial cases.

Formal architectural specification facing domain models: Achieving ML robustness by using formal methods in validating ML design consistency against specifications including quality (e.g., used by DeepMind) needs to consider integrating the domain constraints that are already captured using industrial languages and formats (e.g., Piping and Instrumentation Diagrams P&ID, Control Cause and Effect matrices C&E).

Thus, we need to climb the industrial uphill with innovative solutions for contextualization, domain knowledge integration, simulation-based enrichment, and more.

SAML'21: 1st International Workshop on Software Architecture and Machine Learning
hadil.abukwaik@de.abb.com (H. Abukwaik);
benjamin.kloepper@de.abb.com (B. Kloepper);
reuben.borrison@de.abb.com (R. Borrison);
shrikant.bhat@in.abb.com (S. Bhat)

CEUR Workshop Proceedings (CEUR-WS.org)

[1]Sculley, D. et al. "Hidden technical debt in machine learning systems." Advances in neural information processing systems 2015.

[2]Tang, Y. et al. "An Empirical Study of Refactorings and Technical Debt in Machine Learning Systems." The 43rd International Conference on Software Engineering 2021.