Algorithm Debt in Machine and Deep Learning Systems

Emmanuel Iko-Ojo Simon, ANU School of Computing

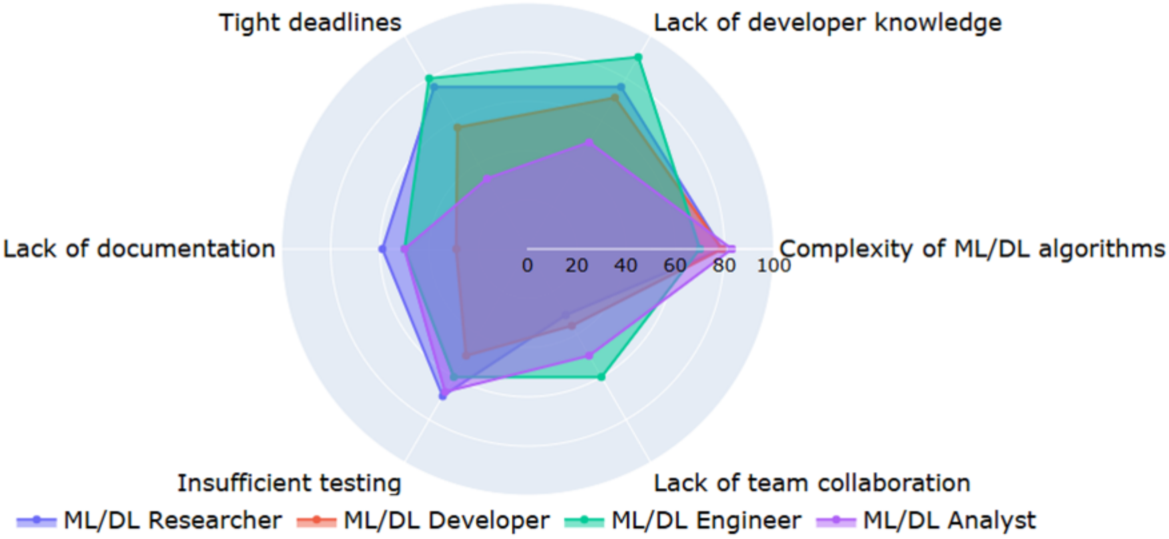

Figure 1: Perceived causes of Algorithm Debt by roles

The graphic (Figure 1), is a radar chart, that visualises the results of a study conducted on the perceived causes of Algorithm Debt in Machine and Deep Learning systems. Algorithm Debt refers to the accumulation of inefficiencies, trade-offs, and overlooked design decisions that can undermine the long-term reliability of Machine and Deep Learning systems. The goal of this study was to investigate how different practitioner roles incur Algorithm Debt to inform mitigation strategies for Algorithm Debt.

65 respondents working in Machine and Deep Learning systems development including Researchers, Developers, Engineers, and Analysts, were surveyed using a questionnaire delivered via ANU Qualtrics. These participants were asked to evaluate six key contributors to AD: algorithmic complexity, lack of developer knowledge, tight deadlines, insufficient documentation, insufficient testing, and lack of team collaboration.

Now, what is displayed in the chart is the result of how each participant grouped by role, rated these factors. From the chart, each color represents a distinct role. Algorithmic complexity stood out across the board, with 83% of Analysts and 79% of Researchers identifying it as a major issue. Engineers, on the other hand, identified knowledge gaps (90%) and deadline pressure (80%) as their top concerns. Developers were more focused on documentation and testing, with only 29% citing documentation. Team collaboration was rated lowest, but still seen as relevant by Engineers and Analysts.

What this means is that AD is not a one-size-fits-all problem, but is affected by role-specific challenges. This radar chart above helps us visualise the differences and points to the need for tailored interventions for each role. These findings led to the development of an integrated framework on AD that was later developed during my thesis, and they overall highlight the importance of managing AD proactively to ensure the reliability of ML/DL systems as they scale.