Research Summary

Safwat Hassan
Assistant Professor
University of Toronto, ON, Canada
safwat.hassan@utoronto.ca

Mobile applications (*apps*) are becoming an integral part of our daily activities. The mobile market is continuously growing, with an expected $156 billion in revenue by 2023 [4]. With the mobile market's highly competitive nature, app developers need to avoid deploying buggy releases and continuously deploy high-quality releases that satisfy user needs. Over the years, the growing mobile apps market with millions of apps and billions of yearly downloads generates comprehensive data. Mining mobile app market data (e.g., analyzing user reviews) can help app developers improve their apps' perceived quality.

My research focuses on applying data mining techniques to the emerging software repositories (e.g., the app market data) to improve the perceived quality of software products. In the last six years, I have been working on providing techniques that can help app developers and store owners improve mobile apps' perceived quality in the following directions.

1. Analyzing Users' Feedback

We proposed approaches that help app developers proactively spot buggy releases and deploy fixes for such buggy releases [20, 32, 25]. Moreover, we proposed approaches to spot user complaints that may lead to user churn (i.e., migrating to competitor products) [14] and prioritize the required features based on the app's competitors [6, 21, 7]. We proposed an approach to identify the topics of the raised issues in the bug reports [5]. We also proposed approaches to help app developers design mobile apps with a high-quality user interface (UI) [11].

2. Analyzing Developers' Common Practices

We studied the common practices of delivering main features in mobile apps, such as the patterns of writing release notes [33], integrating third-party libraries [1, 2, 31, 30, 18], and managing deep learning frameworks [8]. We analyzed the practices of integrating Large Language models (LLMs) in mobile apps [24] and how users perceive the adoption of LLMs in mobile apps [3]. Our work shows the challenges and promises of adopting LLM in mobile apps [3, 24]. We also identified the common mistakes that lead to emergency fixes in mobile apps [22]. This work highlighted the importance of analyzing the store data on a large scale to automatically identify the deployed issues and recommend solutions to the identified issues.

3. Studying the Interactions Between Users and Developers

Responding to user reviews can improve the ratings of an app. However, with many daily user reviews, it is not practical to respond to all reviews. Hence, we proposed an automatic approach to predict reviews that are most likely to get app developers' responses [23]. We also studied developers' interactions using development communication platforms, such as the Gitter platform [13]. We proposed an approach that identifies discussion threads in communication platforms. Project maintainers can benefit from our work to spot the commonly repeated discussions and use such discussions to improve

the documentation of their projects. To ease the replication of our study, we shared our dataset in our replication package [13].

4. Improving the Performance of Software Systems

We collaborate with IBM to provide approaches to enhance the performance of software systems [35, 15, 34]. We also proposed different approaches to improve the design and maintenance of software systems [26, 27, 28].

5. Adopting Mobile Apps for Healthy Life

We collaborated with medical practitioners to build mobile apps that adopt machine learning techniques to predict medical issues in users (patients). In addition, we proposed approaches that provide proactive recommendations for patients and practitioners to avoid severe consequences of the predicted medical issues [12, 29].

6. Towards Rapid Releases of Software Systems

We analyze the build results of 924,616 builds of 588 GitHub projects. We identified best practices to improve the build performance of open-source software systems [17]. We also study the practices of adopting Continuous Integration and Continuous Delivery **(CI/CD)** in mobile apps [16, 9, 36]. In addition, we proposed "LogSieve" as an approach to reduce the generated CI logs in mobile apps [10]. Our work aims to identify the promises and challenges of adopting CI/CD practices in mobile apps [19, 36]. Our work shows variations in the impact of adopting CI practices on the success of mobile apps [36].

References

[1] M. Ahasanuzzaman, S. Hassan, C. Bezemer, and A. E. Hassan. A longitudinal study of popular Ad libraries in the Google Play Store. *Empirical Software Engineering*, 25(1):824–858, 2020.

[2] M. Ahasanuzzaman, S. Hassan, and A. E. Hassan. Studying Ad library integration strategies of top free-to-download apps. *IEEE Transactions on Software Engineering*, pages 1–12, 2020.

[3] B. AlMulla, M. Assi, and S. Hassan. Understanding the challenges and promises of developing generative AI apps: An empirical study. *CoRR*, abs/22506.16453, 2025.

[4] B. O. Apps. App revenue statistics (2019). `https://www.businessofapps.com/data/app-revenues/`. (Last accessed: March 2019).

[5] M. Assi, S. Hassan, S. Georgiou, and Y. Zou. Predicting the change impact of resolving defects by leveraging the topics of issue reports in open source software systems. *ACM Transactions on Software Engineering and Methodology*, 32(6):141:1–141:34, 2023.

[6] M. Assi, S. Hassan, Y. Tian, and Y. Zou. Featcompare: Feature comparison for competing mobile apps leveraging user reviews. *Empirical Software Engineering*, 26(5):94, 2021.

[7] M. Assi, S. Hassan, and Y. Zou. LLM-Cure: LLM-based competitor user review analysis for feature enhancement. *ACM Transactions on Software Engineering and Methodology*, 2025.

[8] M. Assi, S. Hassan, and Y. Zou. Unraveling code clone dynamics in deep learning frameworks. *ACM Transactions on Software Engineering and Methodology*, 2025.

[9] M. E. Barnes, T. A. Ghaleb, and S. Hassan. Task-aware reduction for scalable LLM–database systems. In *Proceedings of the 35th International Conference on Collaborative Advances in Software and Computing Workshop on LLMs Meet Databases: Next-Generation Data Systems*, CASCON '25, 2025.

[10] M. E. Barnes, T. A. Ghaleb, and S. Hassan. Logsieve: Task-aware CI log reduction for sustainable LLM-based analysis. In *Proceedings of the 23rd International Conference on Mining Software Repositories*, MSR '26, pages 1–10. ACM, 2026.

[11] Q. Chen, C. Chen, S. Hassan, Z. Xing, X. Xia, and A. E. Hassan. How should I improve the UI of my app? a study of user reviews of popular apps in the Google Play. *ACM Transactions on Software Engineering and Methodology*, 30(3):1–38, 2021.

[12] N. Dekhil, Y. Mamatjan, S. Hassan, and M. Salih. A novel recommender system for stroke risk stratification. In *Proceedings of the 19th IEEE Conference on Computational Intelligence inBioinformatics and Computational Biology (CIBCB)*, CIBCB '22, pages 1–2, 2022.

[13] O. Ehsan, S. Hassan, M. E. Mezouar, and Y. Zou. An empirical study of developer discussions in the Gitter platform. *ACM Transactions on Software Engineering and Methodology*, 30(1):8:1–8:39, 2021.

[14] O. El Zarif, D. A. da Costa, S. Hassan, and Y. Zou. On the relationship between user churn and software issues. In *Proceedings of the 17th International Conference on Mining Software Repositories*, MSR '20, pages 339–349. ACM, 2020.

[15] O. El Zarif, S. Hassan, Y. Zou, C. Zuzarte, and V. Corvinelli. Pred-cache: a predictive caching method in database systems. In *Proceedings of the 30th Annual International Conference on Computer Science and Software Engineering*, CASCON '20, pages 93–102. ACM, 2020.

[16] T. A. Ghaleb, O. Abduljalil, and S. Hassan. CI/CD configuration practices in open-source Android apps: An empirical study. *ACM Transactions on Software Engineering and Methodology*, 2025.

[17] T. A. Ghaleb, S. Hassan, and Y. Zou. Studying the interplay between the durations and breakages of continuous integration build. *IEEE Transactions on Software Engineering*, 49(4):2476–2497, 2023.

[18] K. A. Hasan, J. Yasmin, H. Hao, Y. Tian, S. Hassan, and S. H. H. Ding. Understanding abandonment and slowdown dynamics in the maven ecosystem. In *Proceedings of the 22nd International Conference on Mining Software Repositories*, MSR '25, pages 1–4. ACM, 2025.

[19] S. Hassan. Opportunities and challenges of adopting CI/CD practices in mobile apps. In *Proceedings of the 35th International Conference on Collaborative Advances in Software and Computing*, CASCON '25, 2025.

[20] S. Hassan, C. Bezemer, and A. E. Hassan. Studying bad updates of top free-to-download apps in the Google Play Store. *IEEE Transactions on Software Engineering*, 46(7):773–793, 2020.

[21] S. Hassan, H. Li, and A. E. Hassan. On the importance of performing app analysis within peer groups. In *Proceedings of the 29th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, SANER '22, pages 1–12, 2022.

[22] S. Hassan, W. Shang, and A. E. Hassan. An empirical study of emergency updates for top Android mobile apps. *Empirical Software Engineering*, 22(1):505–546, 2017.

[23] S. Hassan, C. Tantithamthavorn, C. Bezemer, and A. E. Hassan. Studying the dialogue between users and developers of free apps in the Google Play Store. *Empirical Software Engineering*, 23(3):1275–1312, 2018.

[24] K. Hau, S. Hassan, and S. Zhou. LLMs in mobile apps: Practices, challenges, and opportunities. In *Proceedings of the 12th IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, MOBILESoft '25, pages 1–12, 2025.

[25] H. Hu, Y. Tian, S. Hassan, and D. Lin. Analyzing gamer complaints in reviews of cross-platform video games on steam. In *IEEE Conference on Games, CoG 2023, Boston, MA, USA, August 21-24, 2023*, pages 1–8. IEEE, 2023.

[26] S. M. Ibrahim, S. A. Salem, M. A. Ismail, and M. Eladawy. Identification of nominated classes for software refactoring using object-oriented cohesion metrics. *International Journal of Computer Science Issues (IJCSI)*, 9(2):68, 2012.

[27] S. M. Ibrahim, S. A. Salem, M. A. Ismail, and M. Eladawy. Novel sensitive object-oriented cohesion metric. In *Proceedings of the 22nd International Conference on Computer Theory and Applications (ICCTA)*, ICCTA '12, pages 154–159, 2012.

[28] Y. Jia, S. Hassan, and Y. Zou. Enhancing software maintenance: A learning to rank approach for co-changed method identification. *CoRR*, abs/2411.19099, 2024.

[29] S. Khan, N. Dekhil, E. Mamatjan, S. Hassan, and Y. Mamatjan. An automated online recommender system for stroke risk assessment. In *Proceedings of the 45th Conference of The Canadian Medical and Biological Engineering Society (CMBES)*, CMBES '23, pages 1–4, 2023.

[30] A. K. McIntosh, S. Hassan, and A. Hindle. What can Android mobile app developers do about the energy consumption of machine learning? *Empirical Software Engineering*, 24(2):562–601, 2019.

[31] A. Polese, Y. Tian, and S. Hassan. Adoption of third-party libraries in mobile apps: A case study on open-source Android applications. In *Proceedings of the 9th IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, MOBILESoft '22, pages 1–12, 2022.

[32] I. Saidani, A. Ouni, M. Ahasanuzzaman, S. Hassan, M. W. Mkaouer, and A. E. Hassan. Tracking bad updates in mobile apps: A search-based approach. *Empirical Software Engineering*, pages 1–41, 2022.

[33] A. Z. Yang, S. Hassan, Y. Zou, and A. E. Hassan. An empirical study on release notes patterns of popular apps in the Google Play Store. *Empirical Software Engineering*, pages 1–41, 2021.

[34] G. Zhao, S. Georgiou, Y. Zou, S. Hassan, D. Truong, and T. Corbin. Enhancing performance bug prediction using performance code metrics. In *Proceedings of the 21st International Conference on Mining Software Repositories*, MSR '24, pages 50–62. ACM, 2024.

[35] G. Zhao, S. Hassan, Y. Zou, D. Truong, and T. Corbin. Predicting performance anomalies in software systems at run-time. *ACM Transactions on Software Engineering and Methodology*, 30(3):1–33, 2021.

[36] X. Zhou, T. A. Ghaleb, and S. Hassan. Role of CI adoption in mobile app success: An empirical study of open-source Android projects. In *Proceedings of the 23rd International Conference on Mining Software Repositories*, MSR '26, pages 1–10. ACM, 2026.