

Explainability for Embedding AI: Aspirations and Actuality

Thomas Weber thomas.weber@ifi.lmu.de^[0000-0002-6894-605X]

LMU Munich, Munich, Germany

Abstract. With artificial intelligence (AI) embedded in many everyday software systems, effectively and reliably developing and maintaining AI systems becomes an essential skill for software developers. However, the complexity inherent to AI poses new challenges. Explainable AI (XAI) may allow developers to understand better the systems they build, which, in turn, can help with tasks like debugging. In this paper, we report insights from a series of surveys with software developers that highlight that there is indeed an increased need for explanatory tools to support developers in creating AI systems. However, the feedback also indicates that existing XAI systems still fall short of this aspiration. Thus, we see an unmet need to provide developers with adequate support mechanisms to cope with this complexity so they can embed AI into high-quality software in the future.

Keywords: explainable AI · explanatory debugging · debugging AI · data-driven development.

1 Introduction

In recent years, the proliferation of artificial intelligence (AI) technologies has revolutionized various industries, with practical applications in fields such as medicine [26], marketing [6], IT security [2], and also in everyday life. However, building reliable systems in many of these domains is already challenging. Adding the complexity inherent in AI often makes the software hard to understand for developers and end-users alike. This is particularly the case because, with these types of applications, some critical behavior is no longer encoded by the developer but is instead inferred from data. This makes debugging and maintenance of these systems increasingly challenging.

Explainable AI (XAI) has emerged as a promising domain to address some of these challenges by adding transparency and interpretability to the AI models and their behavior. Ideally, this would mean that developers can use these XAI mechanisms to understand the software they write and any opaque behavior that may be inferred from the data. This might help with debugging and thus enhance the overall robustness and reliability of AI-powered software systems.

However, despite its potential benefits, XAI is not without its limitations. Aside from technical challenges, like negatively affecting performance [4,5], the

interpretability of AI models may vary depending on factors such as model complexity, data heterogeneity, and task characteristics, posing challenges to the generalizability of XAI solutions. Furthermore, how much value an XAI system can provide can be subjective, raising concerns about the benefit across different users and applications. Thus, creating XAI systems not with a technical perspective but relying on human-centered research to determine the actual practical benefits, is critical for their success.

In this paper, we outline some insights we have collected through surveys with software developers regarding XAI systems and how much they actually benefit from them. They highlight that, indeed, XAI system could be a way to support developers but also that current popular XAI systems may still fall short of this aspiration. This emphasizes that there is still a continuing need for human-centric research on XAI methods beyond their initial case studies.

2 Related Work

While embedding AI into software opens up many opportunities for novel applications, it also introduces new challenges. Since in modern AI systems, the developer no longer encodes the behavior of the system, but instead, it is inferred from data, it becomes challenging to understand their behavior, particularly when things do not go as intended.

In response, XAI research tries to make these systems more explainable, interpretable, and understandable [13]. Over the years, people have developed many mechanisms, visualizations, and tools to provide explanations in some form on another [1,3,9,12,15,20]. Some popular examples for these are LIME [25] which uses an explanatory surrogate model or SHAP [21], which utilizes Shapley values to assign importance to input features.

The idea to use explanatory systems for debugging software has also been around for quite some time, initially for traditional software [19], but increasingly also specifically targeting AI systems [18,24] due to their complexity and opacity.

The research community has, however, acknowledged several shortcomings of current XAI systems [7]. For example, while there are many viable use cases, many current XAI systems insufficiently address users' needs in these scenarios. One reason for this is their often prototypical nature, but also the fact that these XAI systems are very narrow in scope and focus only on use cases from their conception [7]. Thus, more human-centric research will be necessary.

3 Surveys

To determine how software developers perceive XAI systems, we collected feedback through three online surveys.

Survey 1: The Need for Explainability In the first survey, our goal was to initially gauge the need for explainability in general. At this point, we considered a broad opinion from developers and end users alike. To this end, we applied

the survey scale described by Weber et al. [28] with a series of different scenarios and types of software. In the initial version of the survey, we included nine applications that embed AI as one part of its functionality. Five were picked to be widely available (online search, social media, multimedia platforms, shopping recommender systems, and navigation), and four were not yet as common (autonomous driving, predictive policing, robotics, personalized medicine). These examples were based on common examples for AI applications from the literature [8,10,11,12,16,23,27].

The survey started with a consent form and questions about the participants' demographic background and technology affinity. Then, each scenario and application was briefly described to participants, after which they answered statements about their understanding of and interest in explanations for this system using five-point Likert scales. To put the demand for explainability of AI systems into context, we later added an additional five applications that do not utilize AI for their core functionality (file management, web browser, email, office software like MS PowerPoint, media manipulation software like Adobe Photoshop). The order of the applications was randomized in the survey.

Survey 2: Explainability for Developers Based on the results of the first survey, we decided to investigate further how XAI methods are perceived by software developers, specifically with the goal of improving the development experience and understanding the systems they implement.

After initial consent and demographic items, we collected feedback on existing XAI systems. We used LIME [25] and SHAP [21] as exemplary systems, as these two are also popular and established examples in the literature [7]. For each of these, participants received a brief explanation and a concrete example using several datasets [14]. After familiarizing themselves with the XAI system, participants responded to a series of statements about it using five-point Likert scales. These statements were based on the survey scales on the demand for explainability from the previous survey. Still, they were extended and rephrased to accommodate the presence of a dedicated explanatory system and focus on the perspective of software developers. After this, we tested the participants' understanding of the presented methods by asking them a series of multiple-choice questions. Each of these was phrased as a potential conclusion that could have been drawn from the XAI system. Participants then needed to decide whether the conclusion was drawn from the XAI system. As before, the order in the survey was randomized.

Survey 3: Explainability for Recognizing Faults Finally, the third survey aimed to see whether XAI allowed developers to draw their own conclusions to detect issues in the system.

To determine whether participants could utilize XAI methods to detect faults, we asked them to use again LIME and SHAP the California Housing Dataset [17]. In this survey, we manipulated the dataset to introduce faults. To simulate skewed data, we replaced the median income in a quarter of the data points where the median house value was above 200.000. Additionally, we swapped the

labels of one of the most important features, the longitude, and one of the least important, the number of households. Since latitude remained an important feature, this led to semantic inconsistencies, as the position of a household consisted of one very important and one seemingly unimportant feature.

Again, we first collected participants’ consent and demographic information. After this, the dataset and the explanatory methods were introduced. We asked participants to investigate whether the system works as intended using LIME and SHAP in separate study conditions. Each participant used the unmodified and the modified dataset. The order of the conditions was randomized. They then answered questions about their experience with the XAI system and the dataset in six categories: (1) ranking of the three most important features, (2) multiple choice questions about the relationship between features and prediction, as well as five-point Likert scales on (3) their subjective understanding of the model, (4) the perceived plausibility of the explanation of the model, and (5) their confidence in the model. Finally, we questioned (6) whether they could detect anomalies or errors in the data or the model.

3.1 Apparatus and Participants

We conducted all surveys online via an institutional survey platform. Participants could take the survey at any time but were required to complete it in one continuous session. The link to the survey and, thus, recruitment was distributed through mailing lists of our institution and external contacts with professional developers. Participants were compensated with the equivalent of 10 \$US per hour.

This recruitment yielded 96 participants for the first survey with just AI-powered systems and 116 who completed the full survey with AI-powered and traditional examples (94 male, 112 female, six other). Half of this sample were students of various fields with an average age of 25 years (SD: 5.7 years). The second survey was completed by 17 participants (12 male, four female, and one other), either Computer Science students or working as data scientists (3), software engineers or architects (5), or in research (1). Finally, the third survey was completed by 21 participants (12 male, 6 female, 3 other). For this survey, we had a majority of responses from Computer Science students (14), while the other seven participants were professional software developers (3), data scientists (2), or academic researchers (2). In consequence half of the participants was younger than 25 while the other half was in the 25–34 year range.

4 Results

In the following, we will summarize highlights from the survey responses.

In the first survey, we observed a generally high demand for explanations for all 14 presented systems. However, taking into account technology affinity, we observed that participants reported low technology affinity had an equally high demand for explainability for AI and non-AI applications. Participants with

higher technology affinity and particularly background in Computer Science had on average a lower demand for explanations for the traditional applications. However, for the AI-powered systems, their demand for explanations was, on average, about the same as that of the inexperienced participants.

As this suggests, even experienced users may require explanations for software with AI functionality about as much as the average user. The goal of our analysis was then to see whether XAI systems can assist them in this situation. However, based on the feedback from the second survey, this does not appear to be the case immediately. In the responses, we observed quite consistently equal groups of those who considered the XAI systems positively, helpful, and supporting their understanding, as well as those who were rather negative and saw no immediate benefit from them. The only area where there was a notable imbalance was regarding trust in the system, where two-thirds of the participants saw no positive impact of the XAI systems for calibrating their trust.

When taking into account experience, this picture shifts, though: for example, participants with multiple years of data science experience were generally more confident in their ability to interpret explanatory output, thus also seeing the greater benefit of understanding the behavior of the AI system. Similarly, the more experienced participants found the XAI systems generally easy to understand while only a single less experienced participant considered them easy to understand.

Also, less experienced participants were all uncertain to sceptical whether LIME or SHAP helps them detect issues with an AI system they are building. Meanwhile, half of the experienced developers saw at least some benefit for finding faults, while the other half was neutral for both XAI systems.

Thus, we used the third survey to test the potential benefit of finding faults. Feedback from this survey indicates that participants had an overall positive impression of the XAI methods for their given task. The perceived understanding, subjective plausibility, and confidence in the model were overall high, with no significant differences between the two XAI methods. However, only two participants were able to detect one of the deliberately added faults when using SHAP, and only one participant found a fault using LIME.

5 Discussion

Considering the feedback from these surveys, it is quite clear that developers are a viable target group for XAI. For end-users, explanations may be useful, but that is true, regardless of whether the system has embedded AI functionality. From an end-user’s perspective, the internal mechanisms are opaque either way. A mental model where the presence of AI makes a difference will often require at least some degree of technical expertise.

Software developers bring expertise and typically willingness to engage with the details of AI. Understanding existing traditional systems is already well supported because software developers can benefit from their computational thinking skills [29] to construct an adequate mental model. Additionally, there are

well-established debugging tools, logging output, etc. However, these do not seem to satisfy the needs of experienced users. XAI may be a solution here to support developers.

However, the popular XAI tools we presented in our survey, but also many more, appear to be designed more towards a specific initial use case and less for generally exploring potential faults in an AI system or the underlying data. The skewed responses of the experienced users might also suggest that XAI merely helps support existing suspicions or highlight known potential issues. Experienced users may already know what they might be looking for and use the XAI system to confirm their hypotheses. While this can be useful, care needs to be taken that developers do not over-rely on XAI just to support their existing beliefs. The fact that XAI systems seem to be built only for the experts themselves has also been criticized before [22] and means that novice users have an even harder time figuring out faults in their systems.

Even so, the actual rate at which participants could detect deliberately added faults was not great either. One can imagine that even more obscure or situational errors are even harder to detect. However, it must be noted that the XAI systems in our study were not designed as dedicated debugging tools but to get a general understanding of the AI models. At the same time, though, the example use cases we presented were not particularly complex either, and still, the participants had trouble finding the added faults. The participant sample with many inexperienced users may also be a reason for this. However, this only further emphasizes the previous point that these tools require prior knowledge to be useful in the first place.

6 Conclusion

To ensure that software with embedded AI functionality is reliable and generally of high quality it is essential that software developers have appropriate tools to understand what is going on. XAI has the potential to be one such tool that assists developers in understanding and debugging their applications. However, popular XAI systems are not necessarily engineered with such a goal in mind. As feedback from our surveys suggests, they currently require a good deal of prior knowledge to be beneficial, and even then, it is unclear whether they simply support existing mental models or actually provide a broader benefit for finding unknown issues in AI systems.

The feedback underlines that even for experienced developers AI systems pose a challenge and how limited the tool support still is. At the same time, developers are a target group that could benefit particularly much from XAI systems. This makes XAI for developers a particularly interesting research area that can help developers and, by extension, improve the quality of AI applications in general.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* **6**, 52138–52160 (2018)

2. Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A.V., Di Franco, F.: The role of machine learning in cybersecurity. *Digital Threats: Research and Practice* **4**(1), 1–38 (Mar 2023). <https://doi.org/10.1145/3545574>
3. Barbalau, A., Cosma, A., Ionescu, R.T., Popescu, M.: A generic and model-agnostic exemplar synthetization framework for explainable ai (2020)
4. Berk, R., Heidari, H., Jabbari, S., Kearns, M., Roth, A.: Fairness in criminal justice risk assessments: The state of the art (2017)
5. Biswas, S., Rajan, H.: Do the machine learning models on a crowd sourced platform exhibit bias? an empirical study on model fairness. In: *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. p. 642–653. ESEC/FSE 2020, Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3368089.3409704>, <https://doi.org/10.1145/3368089.3409704>
6. Brei, V.: Machine learning in marketing: Overview, learning strategies, applications, and future developments. *Foundations and Trends in Marketing* **14**, 173–236 (01 2020). <https://doi.org/10.1561/17000000065>
7. Brennen, A.: What do people really want when they say they want "explainable ai?" we asked 60 stakeholders. In: *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. p. 1–7. CHI EA '20, Association for Computing Machinery, New York, NY, USA (2020). <https://doi.org/10.1145/3334480.3383047>, <https://doi.org/10.1145/3334480.3383047>
8. Cohen, I.G., Graver, H.: A doctor's touch: What big data in health care can teach us about predictive policing. *SSRN Electronic Journal* (2019)
9. Du, M., Liu, N., Hu, X.: Techniques for interpretable machine learning. *Commun. ACM* **63**(1), 68–77 (Dec 2019)
10. Eiband, M., Völkel, S.T., Buschek, D., Cook, S., Hussmann, H.: When people and algorithms meet: user-reported problems in intelligent everyday applications. In: Fu, W., Pan, S., Brdiczka, O., Chau, P., Calvary, G. (eds.) *Proceedings of the 24th International Conference on Intelligent User Interfaces, IUI 2019, Marina del Ray, CA, USA, March 17-20, 2019*. pp. 96–106. ACM (2019)
11. Gade, K., Geyik, S.C., Kenthapadi, K., Mithal, V., Taly, A.: Explainable ai in industry. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. p. 3203–3204. KDD '19, Association for Computing Machinery, New York, NY, USA (2019)
12. Goebel, R., Chander, A., Holzinger, K., Lecue, F., Akata, Z., Stumpf, S., Kieseberg, P., Holzinger, A.: Explainable ai: The new 42? In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) *Machine Learning and Knowledge Extraction*. pp. 295–303. Springer International Publishing, Cham (2018)
13. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.: XAI - explainable artificial intelligence. *Sci. Robotics* **4**(37) (2019)
14. Harrison, D., Rubinfeld, D.L.: Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management* **5**(1), 81–102 (1978). [https://doi.org/https://doi.org/10.1016/0095-0696\(78\)90006-2](https://doi.org/https://doi.org/10.1016/0095-0696(78)90006-2), <https://www.sciencedirect.com/science/article/pii/0095069678900062>
15. Hoffman, R.R., Mueller, S.T., Klein, G., Litman, J.: Metrics for explainable AI: challenges and prospects. *CoRR* **abs/1812.04608** (2018)
16. Holzinger, A., Biemann, C., Pattichis, C.S., Kell, D.B.: What do we need to build explainable AI systems for the medical domain? *CoRR* **abs/1712.09923** (2017)

17. Kelley Pace, R., Barry, R.: Sparse spatial autoregressions. *Statistics & Probability Letters* **33**(3), 291–297 (1997). [https://doi.org/https://doi.org/10.1016/S0167-7152\(96\)00140-X](https://doi.org/https://doi.org/10.1016/S0167-7152(96)00140-X), <https://www.sciencedirect.com/science/article/pii/S016771529600140X>
18. Kulesza, T., Burnett, M., Wong, W.K., Stumpf, S.: Principles of explanatory debugging to personalize interactive machine learning. In: *Proceedings of the 20th International Conference on Intelligent User Interfaces*. p. 126–137. IUI '15, Association for Computing Machinery, New York, NY, USA (2015). <https://doi.org/10.1145/2678025.2701399>, <https://doi.org/10.1145/2678025.2701399>
19. Kulesza, T., Stumpf, S., Burnett, M., Wong, W.K., Riche, Y., Moore, T., Oberst, I., Shinsel, A., McIntosh, K.: Explanatory debugging: Supporting end-user debugging of machine-learned programs. In: *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*. pp. 41–48 (2010). <https://doi.org/10.1109/VLHCC.2010.15>
20. Lundberg, S.M., Erion, G.G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.: Explainable AI for trees: From local explanations to global understanding. *CoRR* **abs/1905.04610** (2019)
21. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. *CoRR* **abs/1705.07874** (2017), <http://arxiv.org/abs/1705.07874>
22. Miller, T., Howe, P., Sonenberg, L.: Explainable ai: Beware of inmates running the asylum or: How i learnt to stop worrying and love the social and behavioural sciences (2017)
23. Mittelstadt, B.D., Floridi, L.: Transparent, explainable, and accountable AI for robotics. *Sci. Robotics* **2**(6) (2017)
24. Rasouli, P., Yu, I.C.: Explainable debugger for black-box machine learning models. In: *2021 International Joint Conference on Neural Networks (IJCNN)*. pp. 1–10 (2021). <https://doi.org/10.1109/IJCNN52387.2021.9533944>
25. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should i trust you?": Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. p. 1135–1144. KDD '16, Association for Computing Machinery, New York, NY, USA (2016). <https://doi.org/10.1145/2939672.2939778>, <https://doi.org/10.1145/2939672.2939778>
26. Shehab, M., Abualigah, L., Shambour, Q., Abu-Hashem, M.A., Shambour, M.K.Y., Alsalibi, A.I., Gandomi, A.H.: Machine learning in medical applications: A review of state-of-the-art methods. *Computers in Biology and Medicine* **145**, 105458 (2022). <https://doi.org/https://doi.org/10.1016/j.combiomed.2022.105458>
27. Wang, D., Yang, Q., Abdul, A.M., Lim, B.Y.: Designing theory-driven user-centric explainable AI. In: Brewster, S.A., Fitzpatrick, G., Cox, A.L., Kostakos, V. (eds.) *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*. p. 601. ACM (2019)
28. Weber, T., Hußmann, H., Eiband, M.: Quantifying the demand for explainability. In: Ardito, C., Lanzilotti, R., Malizia, A., Petrie, H., Piccinno, A., Desolda, G., Inkpen, K. (eds.) *Human-Computer Interaction – INTERACT 2021*. pp. 652–661. Springer International Publishing, Cham (2021)
29. Wing, J.M.: Computational thinking and thinking about computing. In: *22nd IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2008, Miami, Florida USA, April 14-18,*

2008. p. 1. IEEE (2008). <https://doi.org/10.1109/IPDPS.2008.4536091>,
<https://doi.org/10.1109/IPDPS.2008.4536091>

