Lessons from Usable ML Deployments and Application to Wind Turbine Monitoring

Alexandra Zytek
MIT
Cambridge, MA, USA
zyteka@mit.edu

Wei-En Wang
MIT
weinwang@mit.edu

Sofia Koukoura
ScottishPower Renewables
Glasgow, Scotland, UK
skoukoura@scottishpower.com

Kalyan Veeramachaneni
MIT
kalyan@csail.mit.edu

Abstract

Through past experiences deploying what we call *usable ML* (one step beyond explainable ML, including both explanations and other augmenting information) to real-world domains, we have learned three key lessons. First, many organizations are beginning to hire people who we call "bridges" because they bridge the gap between ML developers and domain experts, and these people fill a valuable role in developing usable ML applications. Second, a configurable system that enables easily iterating on usable ML interfaces during collaborations with bridges is key. Finally, there is a need for continuous, in-deployment evaluations to quantify the real-world impact of usable ML. Throughout this paper, we apply these lessons to the task of wind turbine monitoring, an essential task in the renewable energy domain. Turbine engineers and data analysts must decide whether to perform costly in-person investigations on turbines to prevent potential cases of brakepad failure, and well-tuned usable ML interfaces can aid with this decision-making process. Through the applications of our lessons to this task, we hope to demonstrate the potential real-world impact of usable ML in the renewable energy domain.

1 Introduction

Over the past few years, we have been developing a system we call Sibyl that will support the use of machine learning (ML) model outputs in decision making even for those who are not ML experts. Systems like Sibyl consist of an ML model combined with the ability to configure ML explanations and interfaces tailored to augment decision-making workflows. We use the term *usable* ML in addition to the more commonly used term *explainable* AI (XAI) because ML model outputs may require more than just explanations in order to be used effectively for decision making, and fully explained ML models may nonetheless be difficult to use [12, 17, 24]. Configurability, continuous evaluation mechanisms, and iterative updates in collaboration with key people are needed to develop effective usable ML interfaces.

We have learned that deploying ML model outputs in a real scenario — in other words, with people working on the decision problem the model outputs are being used to solve — is the only way to get good feedback. Relying on toy datasets (however realistic) and formal user studies (however carefully chosen) cannot provide feedback with the necessary depth. To this end, when working to deploy ML model outputs, we seek close collaborations with the people who are

XAI in Action: Past, Present, and Future Applications @ NeurIPS 2023.

actually using them. As of when this paper was written, we have worked on two such real-world deployments in two different domains: child welfare screening and wind turbine monitoring. In this paper, we focus on the latter deployment, a current work-in-progress, and share lessons we believe will apply to ML deployments in general.

Our case study. To keep turbines running effectively, operators analyze data to determine when a potential failure may occur, in order to avoid unnecessary costs and downtime. One type of failure is when a turbine brakepad prematurely wears out. This kind of failure can be prevented by sending technicians up the turbines for investigation and repair, but this is an expensive and potentially dangerous task. A deployed usable ML application could reduce downtime from such failures by alerting the relevant personnel to potential brakepad failures and provide information that enables them to make efficient decisions about brakepad replacement.

In order to develop an effective usable ML application for this problem, our team is working in parallel on the two parts of this application: the ML model development and explanations/augmenting interfaces for the ML model output. This paper focuses on the latter task. The ML model is an XGBoost classifier [5] that predicts whether or not a brakepad is likely to fail in a given time window and uses around 1,400 features to do so. The features include readings of the turbine, such as temperatures of components and vibration data.

By combining this experience with our previous usable ML application in the domain of child welfare screening, a project that has lasted several years [24], we have been able to synthesize important transferable lessons. These are:

A new role is emerging: "Bridges". Highlighting the various roles people play in deploying ML model outputs (Section 2), we recognized that developing and evaluating usable ML interfaces are context- and domain-dependent tasks that require collaboration with the right group of people within the domain at hand [3, 10, 11]. We found that a new role is emerging and rapidly gaining traction - that of people within companies who are tasked with connecting domain experts with ML developers. We refer to people in this position as "bridges".

An easy-to-configure system for developing usable ML interfaces is key. Next, in Section 3, we discuss our system Sibyl. To aid with the process of developing and tuning usable ML interfaces for specific domains, we have developed a generalizable system called Sibyl. Sibyl includes a Python library for generating understandable ML explanations, a generalizable back-end layer accessed through a REST-API, and a "lightweight" front-end application built with Streamlit that can be easily adapted for use in new domains. With this system, we can abstract out common overhead code to focus on configuring usable ML interfaces to specific domains.

Continuous evaluation and crafting KPIs is essential. Finally, in Section 4, we discuss the process of evaluating usable ML applications. Evaluating usable ML applications and XAI is a notoriously difficult task due to the complexity of real-world domains [10, 14, 19, 22]. We identified through our past experiences that formal user studies fall short in assessing the real-world impact of ML, and are often too time-consuming for users. We therefore devised an evaluation plan built on tracking existing key performance indicators (KPIs) through a live deployment.

Through our improved understanding of the key roles, systems, and evaluation processes needed to deploy usable ML in real-world domains, we hope to demonstrate ML's positive impact on this decision-making problem, which can improve the efficiency of wind turbines.

2 Lesson 1: A new role is emerging: "Bridges"

The literature has defined a comprehensive set of people involved in XAI deployment. These include developers who make ML models, ethicists who review the fairness and transparency of ML models, users who use the ML model outputs to make decisions, and affected parties who are impacted by decisions made using ML model outputs [3, 4, 13, 18].

In our previous deployment of a usable ML application for child welfare screening, we aimed to collaborate directly with users (child welfare call screeners), as they are the ultimate audience for the usable ML interfaces. We still believe users are essential to consider, but our experiences have revealed practical issues with this approach. Users have their own jobs to do and often have limited time to offer feedback and participate in evaluations. They generally lack experience with

ML, which can make it difficult for them to identify what explanations and interfaces would be most helpful for them. At the same time, often we (the ML developers) lack the right domain expertise to work directly with users. Each domain has its own intricate issues, workflows, and language which are hard for us to master.

Luckily, many domains already have people who are well-positioned to bridge this gap between ML developers and domain experts. Depending on the domain, their job title may vary. In this paper, we will refer to these people more generally as *bridges*, as they bridge the gap between ML and the domain at hand. Figure 1 summarizes this process. Bridges may or may not be technical experts in ML, but they do have an understanding of how ML is used in their domain, as well as its potential benefits and drawbacks. Additionally, they often act as test users before usable ML interfaces are put in front of users. This makes it easier for them to imagine and suggest potentially helpful changes, and to offer concrete feedback.

The jobs of bridges already involve working with both ML developers and domain experts, and helping to vet and tune ML models for use within their domains; therefore, they are already familiar with the ML evaluation processes used in the domain, which can be adapted to evaluate usable ML interfaces as well. In the world of software development, there is an analogous role — that of the product manager. Product managers bridge the gap between the needs of the end consumers (via collecting feedback from them and creating product requirement documents) and the core software development team. In the child welfare domain, we worked with social scientists, who understand ML development and deployment as well as the child welfare domain and all its associated intricacies. While we interacted with users (child welfare call screeners) directly, these interactions were mediated and organized by the social scientists - the *bridges*.

Bridges in wind turbine monitoring: In the wind turbine domain, the Monitoring and Analysis (M&A) team fills our defined bridge role, specializing in ML/data science applied to wind turbine monitoring. This team kicks off the decision-making process at hand by identifying a problem in the live data with help from the ML model. They then compile a summary of relevant information and visualizations about the issue (for example, "turbine 50's brakepad is predicted to fail because of an increase in the brake caliper temperature"). This summary includes the usable ML interfaces that we are developing, as discussed in the next section.

The M&A team communicates their findings with the Operations and Maintenance (O&M) team, providing them with the compiled summary and explanations. The O&M team, the main users of the interface, then look through this information and make a decision about how to proceed.

If the issue is of significant risk, the O&M team informs the site teams at the wind farm(s) in question about the issue and the suspected cause. The site teams may either fill the role of user (if they also review the model prediction and usable ML interfaces) or affected party (if they carry out the O&M team's suggestions directly). They look into the issue on-site, potentially reaching out to a contracted party to handle repairs.

3 Lesson 2: An easy-to-configure system for developing usable ML interfaces is key

As we work with our collaborators who take on the bridge role (the M&A team) to develop a usable ML interface for brakepad failure prediction, we are going through multiple iterations of the interface design.

This process has reinforced our belief that there is no "one size fits all" when it comes to usable ML interfaces, and therefore a system that makes this iteration process easier is needed. The wind turbine monitoring use case put our system *Sibyl* — a generalizable system to enable the development of usable ML interfaces — to the test. The Sibyl system has three parts. The **Pyreal** library, implemented in Python, generates a variety of ML explanations in an immediately interpretable form. **Sibyl-API**, a back-end REST-API, connects Pyreal to front-end applications. Sibyl-API enables future developers working in different domains to easily configure explanations for their own front-end. Finally, **Sibylapp** is a front-end application for explaining and augmenting ML model outputs[1].

[1]Code and documentation for Sibyl can be found at https://github.com/sibyl-dev

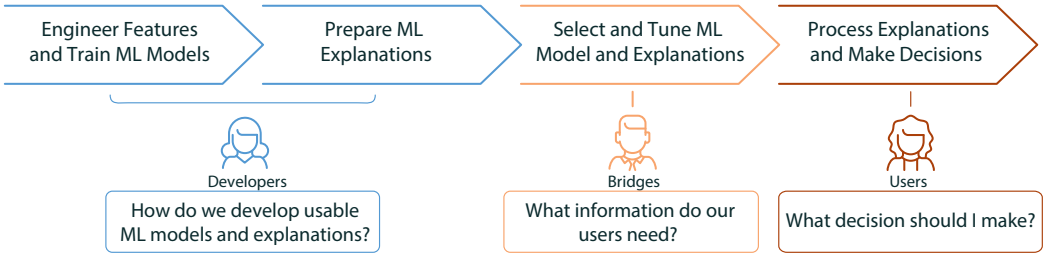

Figure 1: Key roles in usable ML deployment. A new role is emerging, of individuals tasked with bridging the gap between ML developers and users. Bridges enable smoother collaborations throughout the multiple iterations of development required for a usable ML interface.

Sibyl enables making interfaces that show ML explanations and other augmenting information in formats that are readily interpretable and understandable. Interpretable explanations avoid using confounding ML transformations [12, 23], avoid overloading users with information [2, 6, 7], and use positive framing [9].

Since multiple iterations are required to make an effective usable ML interface, we modified Sibylapp's UI from a complex React-based one that required special front-end expertise to Streamlit [1], which allows for easy modification to incorporate feedback. This simple change enables us to iterate faster and creates a lightweight front-end integration.

Configuring usable ML interfaces for turbine brakepad monitoring: For the turbine brakepad monitoring use case, we are iterating on five interfaces, which we summarize briefly here. We have chosen these interfaces based either on previous findings or on direct requests from collaborators. We will add or remove interfaces as needed as we receive further feedback.

***Explore a Prediction*: Local Feature Contributions.** Our first explanatory interface shows the relative positive or negative contribution each feature has made to the model output, calculated using the SHAP algorithm [15]. This interface was chosen because it was found to be the most useful in multiple past investigations [24, 21]. A section of this interface is shown in Figure 2

***Similar Turbines*: Nearest Training-Set Neighbors.** Per requests from collaborators, and based on past findings of usefulness [20], our next interface shows information about the most similar turbine readings from the historic dataset and their outcomes. This page helps users leverage information about past cases that may be relevant to the current scenario. We will work with our collaborators to tune the distance function so we find the most useful similar turbines.

***Compare Timeframes/Turbines*: Explaining Change over Time.** Our next interface allows users to compare a turbine's features, the model prediction, and the model explanation at multiple time points. This interface will allow users to track what changed between "normal" and "failure" predictions, and understand which features specifically contributed to the change. For example, users may see that the brake caliper temperature value decreased, and that the contribution of this feature to the model's prediction increased significantly. This suggests that the temperature change may be relevant to any changes in prediction.

***Understand the Model*: Global Explanations.** In addition to understanding specific alerts, users want to understand the broad trends of turbines so they can make long-term improvements. Our next interface includes several explanation types for this purpose. The first is the feature importance interface, which shows the overall relative importance of each feature to the model's predictions, computed using XGBoost's gain algorithm. Past research [4] has suggested this is one of the most popular explanation types among users. Per feedback from collaborators, we also added an importance metric that retains information about whether a feature contributes positively or negatively, using SHAP. Feature importance can help users better understand the physics of the problem, such as when a failure mode in the brakepad is strongly linked with a specific feature.

***Explore a Feature*: Feature-Level Plots.** We also offer users a way to investigate the effects of specific features on the model prediction, using two types of plots. The first is a scatter plot that includes one point for each row in the database. The x-axis represents each row's value for

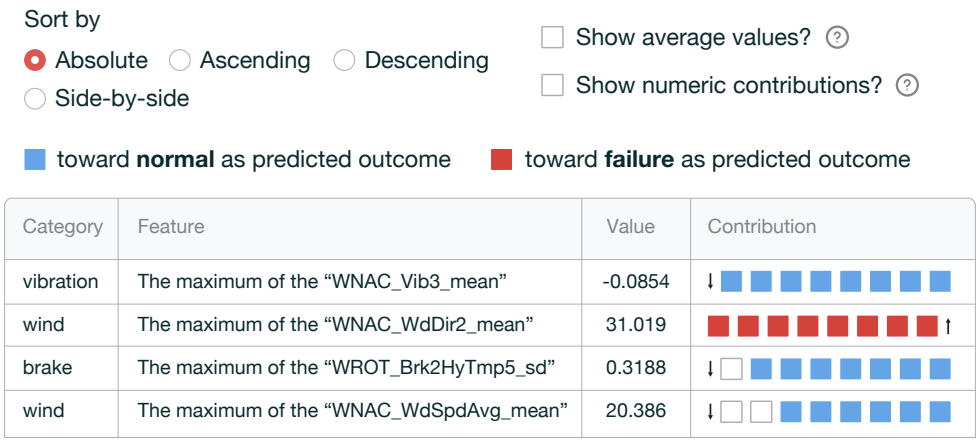

Figure 2: Snippet from the *Explore a Prediction* Sibylapp interface. All Sibylapp interfaces enable users to sort and filter through features by name or category, and offer dynamic sorting options where applicable. On this page, we see the set of features (described in a language meaningful to our end users) that most significantly contributed to the model's final prediction.

the selected feature, while the y-axis shows that feature's contribution for the model's prediction on that row. This can show trends in how the model uses individual features, which users can investigate once they have used other interfaces to identify features of interest. An example of one of these plots is shown in Figure 3. The second plot is a value-distribution plot. Using a box-and-whiskers plot, this shows the minimum, maximum, median, and quartile values of the feature across the dataset, allowing users to quickly understand how the feature is distributed.

As we develop this usable ML application, we continue to identify which interactions between interfaces improve usability. Allowing different explanations to be used together through well-planned interactions is essential to Sibyl's efficient use. For example, users can select a point on the feature-level scatter plot to pull up the full set of feature contributions for that row in the database. In the reverse direction, we plan to enable users to reveal feature-level explanations by selecting rows on the feature contribution or importance tables. This will allow users to efficiently switch between specific cases and the broader context.

4 Lesson 3: Continuous evaluation and crafting KPIs is essential

In complex domains, evaluating the real-world impact of usable ML interfaces and XAI is a challenge. Past work [8, 22] separates evaluations into application-grounded, human-grounded, and functionality-grounded approaches. When working within a specific domain, application-grounded approaches best represent the real-world impact of explainability. Markus et. al. [16] builds on this by distinguishing between empirical and axiomatic evaluation, where the former evaluates a specific metric and the latter evaluates the broader impact on the real-world domain.

We learned from our past study in child welfare that formal user studies (empirical evaluation), while valuable for gaining general knowledge of a field, may be ineffective for evaluating the real-world benefits of specific deployments. User studies held in a lab setting require additional time and attention from users — time that often must be given outside of work hours. Additionally, formal user studies cannot capture the full spectrum of complexity involved in real-world decision-making. Therefore, we aim to evaluate the system with an axiomatic live-deployment approach. A continuous evaluation with live deployment is perhaps the only way to gauge whether a usable ML interface is making a difference in the end goal. This process requires identifying the key performance indicators (KPIs) in the domain — a task bridges are well suited to help with.

Evaluation of the turbine breakpad monitoring interfaces: We are iterating on our usable ML interfaces until we have a version approved by the M&A team who play the bridge role. They will vet the system to ensure it meets the required quality threshold, using the company's existing evaluation processes for new tools.

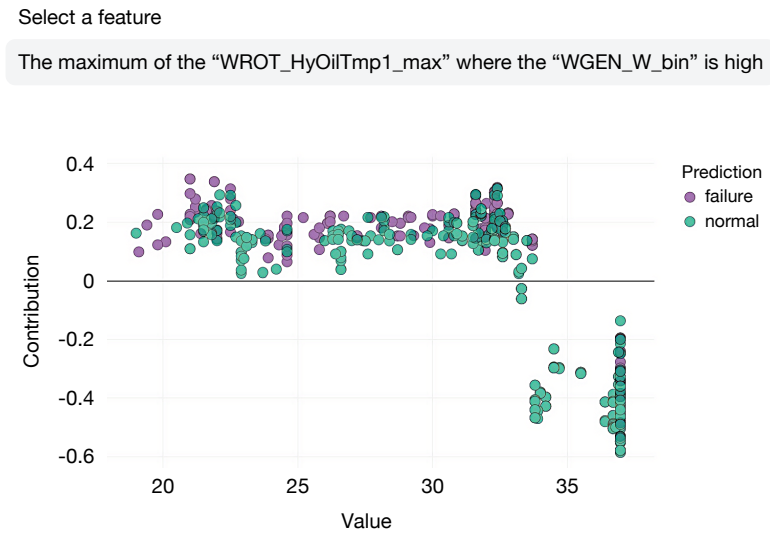

Figure 3: Example of an explanation from the *Explore-a-Feature* interface. This interface generally demonstrates how the model uses a feature, allowing users to dive deeper into feature contributions.

Broadly, our goal is to improve the efficiency of wind turbines, but quantifying this metric requires identifying more specific key performance indicators (KPIs). We have begun identifying existing KPIs for the decision-making problem (preventing brakepad failure). We will then track the shift in these KPIs with the deployment of Sibyl. A few examples of potential KPIs may include 1) the total downtime of turbines during a set time frame, 2) the number of brakepad failures that occurred during a set time frame, compared to the number of in-person investigations performed, and 3) the portion of alerts sent to the O&M team that are further investigated.

Selecting the right KPIs requires choosing metrics as close as possible to the bottom-line company goal (to improve turbine efficiency) while also ensuring they are practical to track [17]. For example, our ML model will improve turbine efficiency chiefly by minimizing downtime and reducing maintenance costs, so Option 1 may be a good choice. However, this metric encompasses so many factors that it may be difficult to isolate the real effects of the introduced usable ML system. Option 2 also captures the benefit of the system; however, actual brakepad failures are uncommon (around one occurs per month across all turbines) so it may not be possible to achieve sufficient statistical power during a practical length of evaluation time. Option 3 strikes a promising balance, capturing the quality of decisions while still being practical to track. We will continue considering other options until the evaluation begins.

Once the KPIs are chosen, we will collect data for one to three months. We will then compare the KPI metrics computed to several historic time frames of the same length, chosen for their similar conditions to the evaluation time frame. This method requires little additional effort from our users beyond performing their usual jobs, and aligns with the existing tool-vetting system used by the company.

5 Conclusion

We are working to deploy and evaluate usable ML interfaces for wind turbine monitoring, using three key lessons from our past experiences. We have identified the team that fills the bridge role by interfacing between ML development and the domain of turbine monitoring — the Monitoring and Analysis team. Through collaborations with this team, we are using our system for usable ML, called Sibyl, to develop appropriate usable ML interfaces for the problem at hand. We are planning on executing a continuous evaluation based on tracking KPIs after deploying the usable ML interfaces to the decision-making process. By taking these steps carefully, we can improve the effectiveness of wind turbines and offer support for the renewable energy industry as a whole.

Acknowledgments and Disclosure of Funding

We thank Iberdrola for funding and data for this project. We thank Robert Jones at ScottishPower Renewables for providing input and feedback on the usable ML system, as well as domain insights about the wind turbine monitoring case study. We would also like to thank Laure Berti-Équille for feedback on our draft, Arash Akhgari for work on our graphics and visualizations, and Cara Giaimo for feedback on writing. Finally, we would like to thank our anonymous reviewers for their insights and feedback.

References

[1] Streamlit: A faster way to build and share data apps, Jan. 2021.

[2] A. Abdul, C. Von Der Weth, M. Kankanhalli, and B. Y. Lim. COGAM: Measuring and Moderating Cognitive Load in Machine Learning Model Explanations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–14, Honolulu HI USA, Apr. 2020. ACM.

[3] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, June 2020.

[4] U. Bhatt, A. Xiang, S. Sharma, A. Weller, A. Taly, Y. Jia, J. Ghosh, R. Puri, J. M. F. Moura, and P. Eckersley. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT* '20, pages 648–657, New York, NY, USA, Jan. 2020. Association for Computing Machinery.

[5] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM. Number of pages: 10 Place: San Francisco, California, USA tex.acmid: 2939785.

[6] H.-F. Cheng, R. Wang, Z. Zhang, F. O'Connell, T. Gray, F. M. Harper, and H. Zhu. Explaining Decision-Making Algorithms through UI: Strategies to Help Non-Expert Stakeholders. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, Glasgow Scotland Uk, May 2019. ACM.

[7] J. Colin, T. Fel, R. Cadène, and T. Serre. What I Cannot Predict, I Do Not Understand: A Human-Centered Evaluation Framework for Explainability Methods. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, Jan. 2023.

[8] F. Doshi-Velez and B. Kim. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv:1702.08608 [cs, stat]*, Mar. 2017. arXiv: 1702.08608.

[9] S. Hadash, M. C. Willemsen, C. Snijders, and W. A. IJsselsteijn. Improving understandability of feature contributions in model-agnostic explainable AI tools. In *CHI Conference on Human Factors in Computing Systems*, pages 1–9, New Orleans LA USA, Apr. 2022. ACM.

[10] P. Hase and M. Bansal. Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior? *Association for Computational Linguistics*, 58:5540–5552, May 2020. arXiv: 2005.01831 version: 1.

[11] S. R. Hong, J. Hullman, and E. Bertini. Human Factors in Model Interpretability: Industry Practices, Challenges, and Needs. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–26, May 2020.

[12] H. Jiang and E. Senge. On Two XAI Cultures: A Case Study of Non-technical Explanations in Deployed AI System. In *Human Centered AI (HCAI) workshop at NeurIPS 2021*, Dec. 2021. arXiv:2112.01016 [cs].

[13] M. Langer, D. Oster, T. Speith, H. Hermanns, L. Kästner, E. Schmidt, A. Sesing, and K. Baum. What Do We Want From Explainable Artificial Intelligence (XAI)? – A Stakeholder Perspective on XAI and a Conceptual Model Guiding Interdisciplinary XAI Research. *Artificial Intelligence*, 296:103473, July 2021. arXiv:2102.07817 [cs].

[14] P. Lopes, E. Silva, C. Braga, T. Oliveira, and L. Rosado. XAI Systems Evaluation: A Review of Human and Computer-Centred Methods. *Applied Sciences*, 12(19):9423, Sept. 2022.

[15] S. M. Lundberg and S.-I. Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 31, page 10, Long Beach, California, 2017. Curran Associates Inc.

[16] A. F. Markus, J. A. Kors, and P. R. Rijnbeek. The role of explainability in creating trustworthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of Biomedical Informatics*, 113:103655, Jan. 2021.

[17] M. Nyre-Yu, E. Morris, B. Moss, C. Smutz, and M. Smith. Considerations for Deploying xAI Tools in the Wild: Lessons Learned from xAI Deployment in a Cybersecurity Operations Setting. In *Proposed for presentation at the ACM SIG Knowledge Discovery and Data Mining Workshop on Responsible AI held August 14-18, 2021 in Singapore, Singapore.* US DOE, May 2021.

[18] A. Preece, D. Harborne, D. Braines, R. Tomsett, and S. Chakraborty. Stakeholders in Explainable AI. In *AAAI FSS-18: Artificial Intelligence in Government and Public Sector*, page 6, Arlington, Virginia, 2018.

[19] A. Rosenfeld. Better Metrics for Evaluating Explainable Artificial Intelligence: Blue Sky Ideas Track. In *Proc. of the 21th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, May 2021.

[20] A. Silva, M. Schrum, E. Hedlund-Botti, N. Gopalan, and M. Gombolay. Explainable Artificial Intelligence: Evaluating the Objective and Subjective Impacts of xAI on Human-Agent Interaction. *International Journal of Human–Computer Interaction*, 39(7):1390–1404, Apr. 2023.

[21] X. Wang and M. Yin. Are Explanations Helpful? A Comparative Study of the Effects of Explanations in AI-Assisted Decision-Making. *26th International Conference on Intelligent User Interfaces*, pages 318–328, Apr. 2021. Conference Name: IUI '21: 26th International Conference on Intelligent User Interfaces ISBN: 9781450380171 Place: College Station TX USA Publisher: ACM.

[22] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger. Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics. *Electronics*, 10(5):593, Jan. 2021. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[23] A. Zytek, I. Arnaldo, D. Liu, and K. Veeramachaneni. The Need for Interpretable Features: Motivation and Taxonomy. *SIGKDD Explorations*, 24(1), June 2022.

[24] A. Zytek, D. Liu, R. Vaithianathan, and K. Veeramachaneni. Sibyl: Understanding and Addressing the Usability Challenges of Machine Learning In High-Stakes Decision Making. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2021. Conference Name: IEEE Transactions on Visualization and Computer Graphics.