# 研究想法

* XAI（Explainable Artificial Intelligence）方法的不一致性问题 XAI disagreement

Roy S, Laberge G, Roy B, et al. Why don’t XAI techniques agree? characterizing the disagreements between post-hoc explanations of defect predictions[C]//2022 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2022: 444-448.

为什么要研究XAI方法的disagreement问题？

（1）信任和可信度：为了使人们信任和接受AI系统，系统的决策需要是可解释的和可理解的。当不同的XAI方法对同一输入产生不一致的解释时，可能会降低人们对AI系统的信任。

（2）决策透明度：在许多应用中，特别是需要对AI系统的决策进行审查和验证的领域，如医疗、司法和金融等，决策的透明度是至关重要的。如果不同的XAI方法给出不同的解释，可能会增加决策的不透明性。

（3）系统健壮性：对XAI方法的不一致性进行研究可以帮助识别和解决系统中可能存在的问题，从而提高系统的健壮性和鲁棒性。

（4）应用广泛性：XAI方法被广泛应用于各种领域，包括医疗、金融、自动驾驶等。了解不同方法之间的不一致性有助于开发更为通用和可靠的XAI解释工具。

（5）进一步的研究和改进：对不同XAI方法之间的不一致性进行研究可以促进对这些方法的改进和优化，从而提高它们的效果和应用价值。

考虑的XAI方法是post-hoc explanation methods

如何度量XAI方法的disagreement？

可以定义一些指标来量化不同XAI方法之间的解释一致性。

解释质量的评估。

Human study

如何提高XAI方法的disagreement？

模型集成：使用模型集成技术，结合多种XAI方法的输出，以生成更为一致和全面的解释。这可以通过加权、投票或其他集成策略来实现。

# 相关论文

[Arxiv 2024] Cao S, Sun X, Widyasari R, et al. A Systematic Literature Review on Explainability for Machine/Deep Learning-based Software Engineering Research[J]. arXiv preprint arXiv:2401.14617, 2024.

基于软工的XAI综述论文

[TSE 2023] Michael Fu, Van Nguyen, Chakkrit Tantithamthavorn, Trung Le, and Dinh Phung. 2023. VulExplainer: A Transformer-Based Hierarchical Distillation for Explaining Vulnerability Types. IEEE Trans. Software Eng. 49, 10 (2023), 4550–4565

[TSE 2023] Shuyao Jiang, Jiacheng Shen, Shengnan Wu, Yu Cai, Yue Yu, and Yangfan Zhou. 2023. Towards Usable Neural Comment Generation via Code-Comment Linkage Interpretation: Method and Empirical Study. IEEE Trans. Software Eng. 49, 4 (2023), 2239–2254

[ICSE 2023]Jiri Gesi, Xinyun Shen, Yunfan Geng, Qihong Chen, and Iftekhar Ahmed. 2023. Leveraging Feature Bias for Scalable Misprediction Explanation of Machine Learning Models. In Proceedings of the 45th IEEE/ACM International Conference on Software Engineering (ICSE). IEEE, 1559–1570.

[ISSTA 2023] Yutao Hu, Suyuan Wang, Wenke Li, Junru Peng, Yueming Wu, Deqing Zou, and Hai Jin. 2023. Interpreters for GNN-Based Vulnerability Detection: Are We There Yet?. In Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA). ACM, 1407–1419.

[ICPC 2023] Mingyang Geng, Shangwen Wang, Dezun Dong, Haotian Wang, Shaomeng Cao, Kechi Zhang, and Zhi Jin. 2023. Interpretation-based Code Summarization. In Proceedings of the 31st IEEE/ACM International Conference on Program Comprehension (ICPC). IEEE, 113–124

[SANER 2023] Gichan Lee and Scott Uk-Jin Lee. 2023. An Empirical Comparison of Model-Agnostic Techniques for Defect Prediction Models. In Proceedings of the 30th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). IEEE, 179–189

[ICSME 2023] Mingwei Liu, Simin Yu, Xin Peng, Xueyin Du, Tianyong Yang, Huanjun Xu, and Gaoyang Zhang. 2023. Knowledge Graph based Explainable Question Retrieval for Programming Tasks. In Proceedings of the 39th IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE, 123–135.

[ICSE 2023] Parvez Mahbub, Ohiduzzaman Shuvo, and Mohammad Masudur Rahman. 2023. Explaining Software Bugs Leveraging Code Structures in Neural Machine Translation. In Proceedings of the 45th IEEE/ACM International Conference on Software Engineering (ICSE). IEEE, 640–652

[FSE 2023] Chao Ni, Xin Yin, Kaiwen Yang, Dehai Zhao, Zhenchang Xing, and Xin Xia. 2023. Distinguishing Look-Alike Innocent and Vulnerable Code by Subtle Semantic Representation Learning and Explanation. In Proceedings of the 31th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE). ACM, 1611–1622

[FSE 2023] Ruoxi Sun, Minhui Xue, Gareth Tyson, Tian Dong, Shaofeng Li, Shuo Wang, Haojin Zhu, Seyit Camtepe, and Surya Nepal. 2023. Mate! Are You Really Aware? An Explainability-Guided Testing Framework for Robustness of Malware Detectors. In Proceedings of the 31th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE). ACM, 1573–1585

[TOSEM 2023] Chong Wang, Xin Peng, Zhenchang Xing, Yue Zhang, Mingwei Liu, Rong Luo, and Xiujie Meng. 2023. XCoS: Explainable Code Search Based on Query Scoping and Knowledge Graph. ACM Trans. Softw. Eng. Methodol. 32, 6 (2023), 140:1–140:28

[TOSEM 2023] ia Yang, Cai Fu, Fengyang Deng, Ming Wen, Xiaowei Guo, and Chuanhao Wan. 2023. Toward Interpretable Graph Tensor Convolution Neural Network for Code Semantics Embedding. ACM Trans. Softw. Eng. Methodol. 32, 5 (2023), 115:1–115:40

[ASE 2023] Jian Zhang, Shangqing Liu, Xu Wang, Tianlin Li, and Yang Liu. 2023. Learning to Locate and Describe Vulnerabilities. In Proceedings of the 38th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 332–344.

[ICSE 2023] Jiayuan Zhou, Michael Pacheco, Jinfu Chen, Xing Hu, Xin Xia, David Lo, and Ahmed E. Hassan. 2023. CoLeFunDa: Explainable Silent Vulnerability Fix Identifcation. In Proceedings of the 45th IEEE/ACM International Conference on Software Engineering (ICSE). IEEE, 2565–2577

[MSR 2022] Yuxiang Gao, Yi Zhu, and Qiao Yu. 2022. Evaluating the eﬀectiveness of local explanation methods on source code-based defect prediction models. In Proceedings of the 19th IEEE/ACM International Conference on Mining Software Repositories (MSR). ACM, 640–645

[IST 2022] Yuxiang Gao, Yi Zhu, and Yu Zhao. 2022. Dealing with imbalanced data for interpretable defect prediction. Inf. Softw. Technol. 151 (2022), 107016

[TSE 2022] Jirayus Jiarpakdee, Chakkrit Kla Tantithamthavorn, Hoa Khanh Dam, and John C. Grundy. 2022. An Empirical Study of Model-Agnostic Techniques for Defect Prediction Models. IEEE Trans. Software Eng. 48, 2 (2022), 166–185.

[FSE 2022] Zeyan Li, Nengwen Zhao, Mingjie Li, Xianglin Lu, Lixin Wang, Dongdong Chang, Xiaohui Nie, Li Cao, Wenchi Zhang, Kaixin Sui, Yanhua Wang,

Xu Du, Guoqiang Duan, and Dan Pei. 2022. Actionable and interpretable fault localization for recurring failures in online service systems. In Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE). ACM, 996–1008

[ISSRE 2022] Yue Liu, Chakkrit Tantithamthavorn, Li Li, and Yepang Liu. 2022. Explainable AI for Android Malware Detection: Towards Understanding Why the Models Perform So Well?. In Proceedings of the IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE). IEEE, 169–180

[ICSE 2022] Yao Wan, Wei Zhao, Hongyu Zhang, Yulei Sui, Guandong Xu, and Hai Jin. 2022. What Do They Capture? - A Structural Analysis of Pre-Trained Language Models for Source Code. In Proceedings of the 44th IEEE/ACM International Conference on Software Engineering (ICSE). ACM, 2377–2388

[TSE 2022] Supatsara Wattanakriengkrai, Patanamon Thongtanunam, Chakkrit Tantithamthavorn, Hideaki Hata, and Kenichi Matsumoto. 2022. Predicting Defective Lines Using a Model-Agnostic Technique. IEEE Trans. Software Eng. 48, 5 (2022), 1480–1496.

[ICPC 2022] Ratnadira Widyasari, Gede Artha Azriadi Prana, Stefanus A. Haryono, Yuan Tian, Hafl Noer Zachiary, and David Lo. 2022. XAI4FL: enhancing spectrum-based fault localization with explainable artifcial intelligence. In Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension (ICPC). ACM, 499–510.

[FSE 2021] Jürgen Cito, Isil Dillig, Seohyun Kim, Vijayaraghavan Murali, and Satish Chandra. 2021. Explaining mispredictions of machine learning models using rule induction. In Proceedings of the 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE). ACM, 716–727

[FSE 2021] Yi Ding, Ahsan Pervaiz, Michael Carbin, and Henry Hoﬀmann. 2021. Generalizable and interpretable learning for confguration extrapolation. In Proceedings of the 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE). ACM, 728–740

[IJICAI 2021] Zhenguang Liu, Peng Qian, Xiang Wang, Lei Zhu, Qinming He, and Shouling Ji. 2021. Smart Contract Vulnerability Detection: From Pure Neural Network to Interpretable Graph Feature and Expert Pattern Fusion. In Proceedings of the 30th International Joint Conference on Artifcial Intelligence (IJCAI). 2751–2759

[ASE 2021] Matteo Paltenghi and Michael Pradel. 2021. Thinking Like a Developer? Comparing the Attention of Humans with Neural Models of Code. In Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 867–879.

[ASE 2021] Chanathip Pornprasit, Chakkrit Tantithamthavorn, Jirayus Jiarpakdee, Michael Fu, and Patanamon Thongtanunam. 2021. PyExplainer: Explaining the Predictions of Just-In-Time Defect Models. In Proceedings of the 36th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 407–418.

[FSE 2021] Md. Rafqul Islam Rabin, Vincent J. Hellendoorn, and Mohammad Amin Alipour. 2021. Understanding neural code intelligence through program simplifcation. In Proceedings of the 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE). ACM, 441–452.

[TOSEM 2021] Bozhi Wu, Sen Chen, Cuiyun Gao, Lingling Fan, Yang Liu, Weiping Wen, and Michael R. Lyu. 2021. Why an Android App Is Classifed as Malware: Toward Malware Classifcation Interpretation. ACM Trans. Softw. Eng. Methodol. 30, 2 (2021), 21:1–21:29

[TOSEM 2021] Deqing Zou, Yawei Zhu, Shouhuai Xu, Zhen Li, Hai Jin, and Hengkai Ye. 2021. Interpreting Deep Learning-based Vulnerability Detector Predictions Based on Heuristic Searching. ACM Trans. Softw. Eng. Methodol. 30, 2 (2021), 23:1–23:31.

[ASE 2020] in Wang, Jin Liu, Li Li, Xiao Chen, Xiao Liu, and Hao Wu. 2020. Detecting and Explaining Self-Admitted Technical Debts with Attention-based Neural Networks. In Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE). IEEE, 871–882

[EMSE 2019] Toshiki Mori and Naoshi Uchihira. 2019. Balancing the trade-oﬀ between accuracy and interpretability in software defect prediction. Empir. Softw. Eng. 24, 2 (2019), 779–825.

[TOSEM 2019] Xiaoxue Ren, Zhenchang Xing, Xin Xia, David Lo, Xinyu Wang, and John Grundy. 2019. Neural Network-based Detection of Self-Admitted Technical Debt: From Performance to Explainability. ACM Trans. Softw. Eng. Methodol. 28, 3 (2019), 15