Supplemental Material for "Top Score on the Wrong Exam: On Benchmarking in Machine Learning for Vulnerability Detection"

NIKLAS RISSE, MPI-SP, Germany
JING LIU, MPI-SP, Germany
MARCEL BÖHME, MPI-SP, Germany

This document contains supplemental data for the paper "Top Score on the Wrong Exam: On Benchmarking in Machine Learning for Vulnerability Detection".

CCS Concepts: • **Security and privacy → Software and application security**; • **Software and its engineering → Software testing and debugging**; • **Computing methodologies → Machine learning**.

Additional Key Words and Phrases: machine learning, vulnerability detection, benchmark, function, LLM, data quality, context, spurious correlations, ML4VD

ACM Reference Format:
Niklas Risse, Jing Liu, and Marcel Böhme. 2025. Supplemental Material for "Top Score on the Wrong Exam: On Benchmarking in Machine Learning for Vulnerability Detection". *Proc. ACM Softw. Eng.* 2, ISSTA, Article ISSTA018 (July 2025), 10 pages. https://doi.org/10.1145/3728887

1 Literature Survey Paper List

Table 1, Table 2, and Table 3 contain the complete lists of papers that we included in our literature survey. Table 1 contains the papers published at the Top-4 Software Engineering conferences (ICSE, ISSTA, ASE, and FSE), Table 2 contains the papers published at the Top-4 Security conferences (USENIX, S&P, CCS, and NDSS), and Table 3 contains the papers published at the Top-4 Software Engineering and Security journals (TSE, TOSEM, TIFS, and TDSC). All papers were published between January 2020 and December 2024. In addition to the 81 papers that match the selection criteria of our literature survey, the tables also contain 28 papers that are either not actually on the topic of software vulnerability detection (column "Vulnerability Detection?" -> No) or do not employ machine learning techniques (column "Machine Learning?" -> No). These papers were excluded from the literature survey, because they do not match the selection criteria, but are still listed here for transparency and completeness.

Authors' Contact Information: Niklas Risse, MPI-SP, Bochum, Germany, niklas.risse@mpi-sp.org; Jing Liu, MPI-SP, Bochum, Germany, jing6@acm.org; Marcel Böhme, MPI-SP, Bochum, Germany, marcel.boehme@acm.org.

ACM 2994-970X/2025/7-ARTISSTA018
https://doi.org/10.1145/3728887

Table 1. Papers from the Top-4 Software Engineering conferences (ICSE, ISSTA, ASE, FSE).

Paper	Venue	Year	Vulnerability Detection?	Machine Learning?	Granularities
[12]	ISSTA	2024	Yes	Yes	function, statement
[75]	ISSTA	2024	Yes	Yes	function
[7]	ISSTA	2024	Yes	Yes	file
[57]	ISSTA	2024	Yes	Yes	function
[53]	ISSTA	2023	Yes	Yes	function, statement
[25]	ISSTA	2023	Yes	Yes	function
[101]	ISSTA	2023	No	N/A	N/A
[17]	ISSTA	2023	Yes	Yes	function
[11]	ISSTA	2022	Yes	Yes	function, inter-procedural slice
[4]	ASE	2024	Yes	Yes	function
[37]	ASE	2024	Yes	Yes	function
[83]	ASE	2024	Yes	Yes	function
[104]	ASE	2024	Yes	Yes	function
[78]	ASE	2023	Yes	Yes	function
[99]	ASE	2023	Yes	Yes	function
[51]	ASE	2023	Yes	Yes	function
[34]	ASE	2023	No	N/A	N/A
[39]	ASE	2023	Yes	Yes	function
[59]	ASE	2023	No	N/A	N/A
[47]	ASE	2021	No	N/A	N/A
[41]	ASE	2020	Yes	Yes	function
[58]	ICSE	2024	Yes	Yes	function
[44]	ICSE	2024	No	N/A	N/A
[62]	ICSE	2024	Yes	Yes	function, inter-procedural slice, line
[3]	ICSE	2024	Yes	Yes	function
[66]	ICSE	2024	Yes	Yes	function
[33]	ICSE	2024	Yes	Yes	function
[45]	ICSE	2024	Yes	Yes	function
[54]	ICSE	2024	Yes	Yes	line
[70]	ICSE	2024	Yes	Yes	function
[106]	ICSE	2024	No	N/A	N/A
[73]	ICSE	2023	Yes	Yes	function
[74]	ICSE	2023	Yes	Yes	function
[67]	ICSE	2023	Yes	Yes	function
[96]	ICSE	2023	Yes	Yes	function
[87]	ICSE	2023	Yes	Yes	function, line
[13]	ICSE	2023	Yes	Yes	function
[84]	ICSE	2022	Yes	Yes	function
[1]	ICSE	2022	Yes	Yes	function, inter-procedural slice, program
[24]	FSE	2024	No	N/A	N/A
[52]	FSE	2023	Yes	Yes	function
[31]	FSE	2023	Yes	Yes	commit
[81]	FSE	2023	Yes	Yes	inter-procedural slice
[103]	FSE	2023	Yes	No	N/A
[60]	FSE	2023	Yes	Yes	function
[21]	FSE	2023	Yes	Yes	function
[20]	FSE	2022	No	N/A	N/A
[30]	FSE	2022	No	N/A	N/A
[29]	FSE	2021	Yes	Yes	function, statement

Table 2. Papers from the Top-4 Security conferences (USENIX, S&P, CCS, NDSS).

Paper	Venue	Year	Vulnerability Detection?	Machine Learning?	Granularities
[69]	S&P	2024	Yes	Yes	file
[18]	USENIX	2024	Yes	Yes	function
[64]	USENIX	2024	Yes	Yes	function, inter-procedural slice, line
[42]	USENIX	2024	Yes	Yes	function, line
[40]	USENIX	2024	No	N/A	N/A
[61]	USENIX	2024	Yes	Yes	function
[65]	USENIX	2024	Yes	Yes	function
[48]	USENIX	2023	Yes	Yes	statement, repository
[79]	USENIX	2023	Yes	No	N/A
[80]	USENIX	2022	Yes	No	N/A
[86]	USENIX	2020	Yes	No	N/A
[26]	CCS	2024	No	N/A	N/A
[92]	CCS	2023	Yes	No	N/A
[46]	NDSS	2023	Yes	Yes	function

Table 3. Papers from top Software Engineering and Security journals (TSE, TOSEM, TIFS, TDSC).

Paper	Venue	Year	Vulnerability Detection?	Machine Learning?	Granularities
[93]	TOSEM	2024	Yes	Yes	function
[105]	TOSEM	2024	No	N/A	N/A
[50]	TOSEM	2024	Yes	Yes	function
[63]	TOSEM	2024	Yes	Yes	function, line
[8]	TOSEM	2024	Yes	Yes	function
[102]	TOSEM	2024	Yes	Yes	function
[68]	TOSEM	2023	Yes	Yes	function
[91]	TOSEM	2023	No	N/A	N/A
[2]	TOSEM	2023	Yes	Yes	function, inter-procedural slice, statement
[98]	TOSEM	2022	No	N/A	N/A
[109]	TOSEM	2021	Yes	Yes	inter-procedural slice
[10]	TOSEM	2021	Yes	Yes	function, inter-procedural slice
[5]	TSE	2024	Yes	Yes	function
[56]	TSE	2024	Yes	Yes	function
[76]	TSE	2024	Yes	Yes	function
[28]	TSE	2024	Yes	Yes	function, line
[90]	TSE	2024	Yes	Yes	function
[77]	TSE	2024	Yes	Yes	function
[89]	TSE	2024	Yes	Yes	function
[94]	TSE	2024	No	N/A	N/A
[55]	TSE	2023	Yes	Yes	function
[97]	TSE	2023	Yes	Yes	function
[100]	TSE	2023	Yes	Yes	function
[19]	TSE	2023	Yes	Yes	function
[14]	TSE	2023	No	N/A	N/A
[32]	TSE	2023	No	N/A	N/A
[72]	TSE	2022	Yes	No	N/A
[6]	TSE	2022	Yes	Yes	function, inter-procedural slice
[22]	TSE	2022	Yes	No	N/A
[16]	TSE	2021	Yes	Yes	file
[95]	TSE	2021	No	No	N/A
[82]	TIFS	2024	Yes	Yes	function, line, statement
[85]	TIFS	2024	Yes	Yes	function, inter-procedural slice
[27]	TIFS	2024	Yes	Yes	function, file
[88]	TIFS	2024	Yes	Yes	function
[71]	TIFS	2021	Yes	Yes	function
[15]	TIFS	2021	Yes	No	N/A
[107]	TDSC	2022	Yes	Yes	function
[9]	TDSC	2024	Yes	Yes	function, inter-procedural slice, statement
[23]	TDSC	2023	No	No	N/A
[36]	TDSC	2022	Yes	Yes	inter-procedural slice
[35]	TDSC	2022	Yes	Yes	inter-procedural slice, line
[43]	TDSC	2022	Yes	Yes	function, program
[49]	TDSC	2022	No	No	N/A
[38]	TDSC	2021	Yes	Yes	function
[108]	TDSC	2021	Yes	Yes	program

References

[1] Sicong Cao, Xiaobing Sun, Lili Bo, Rongxin Wu, Bin Li, and Chuanqi Tao. 2022. MVD: memory-related vulnerability detection based on flow-sensitive graph neural networks. In *Proceedings of the 44th International Conference on Software Engineering* (Pittsburgh, Pennsylvania) *(ICSE '22)*. Association for Computing Machinery, New York, NY, USA, 1456–1468. doi:10.1145/3510003.3510219

[2] Sicong Cao, Xiaobing Sun, Lili Bo, Rongxin Wu, Bin Li, Xiaoxue Wu, Chuanqi Tao, Tao Zhang, and Wei Liu. 2023. Learning to Detect Memory-related Vulnerabilities. *ACM Trans. Softw. Eng. Methodol.* 33, 2, Article 43 (Dec. 2023), 35 pages. doi:10.1145/3624744

[3] Sicong Cao, Xiaobing Sun, Xiaoxue Wu, David Lo, Lili Bo, Bin Li, and Wei Liu. 2024. Coca: Improving and Explaining Graph Neural Network-Based Vulnerability Detection Systems. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 155, 13 pages. doi:10.1145/3597503.3639168

[4] Sicong Cao, Xiaobing Sun, Xiaoxue Wu, David Lo, Lili Bo, Bin Li, Xiaolei Liu, Xingwei Lin, and Wei Liu. 2024. Snopy: Bridging Sample Denoising with Causal Graph Learning for Effective Vulnerability Detection. In *2024 39th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 606–618.

[5] Partha Chakraborty, Krishna Kanth Arumugam, Mahmoud Alfadel, Meiyappan Nagappan, and Shane McIntosh. 2024. Revisiting the Performance of Deep Learning-Based Vulnerability Detection on Realistic Datasets. *IEEE Transactions on Software Engineering* 50, 8 (2024), 2163–2177. doi:10.1109/TSE.2024.3423712

[6] Saikat Chakraborty, Rahul Krishna, Yangruibo Ding, and Baishakhi Ray. 2022. Deep Learning Based Vulnerability Detection: Are We There Yet? *IEEE Transactions on Software Engineering* 48, 9 (2022), 3280–3296. doi:10.1109/TSE.2021.3087402

[7] Yujia Chen, Cuiyun Gao, Zezhou Yang, Hongyu Zhang, and Qing Liao. 2024. Bridge and Hint: Extending Pre-trained Language Models for Long-Range Code. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis* (Vienna, Austria) *(ISSTA 2024)*. Association for Computing Machinery, New York, NY, USA, 274–286. doi:10.1145/3650212.3652127

[8] Baijun Cheng, Shengming Zhao, Kailong Wang, Meizhen Wang, Guangdong Bai, Ruitao Feng, Yao Guo, Lei Ma, and Haoyu Wang. 2024. Beyond Fidelity: Explaining Vulnerability Localization of Learning-Based Detectors. *ACM Trans. Softw. Eng. Methodol.* 33, 5, Article 127 (June 2024), 33 pages. doi:10.1145/3641543

[9] Xiao Cheng, Xu Nie, Ningke Li, Haoyu Wang, Zheng Zheng, and Yulei Sui. 2024. How About Bug-Triggering Paths? - Understanding and Characterizing Learning-Based Vulnerability Detectors. *IEEE Transactions on Dependable and Secure Computing* 21, 2 (2024), 542–558. doi:10.1109/TDSC.2022.3192419

[10] Xiao Cheng, Haoyu Wang, Jiayi Hua, Guoai Xu, and Yulei Sui. 2021. DeepWukong: Statically Detecting Software Vulnerabilities Using Deep Graph Neural Network. *ACM Trans. Softw. Eng. Methodol.* 30, 3, Article 38 (April 2021), 33 pages. doi:10.1145/3436877

[11] Xiao Cheng, Guanqin Zhang, Haoyu Wang, and Yulei Sui. 2022. Path-sensitive code embedding via contrastive learning for software vulnerability detection. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis* (Virtual, South Korea) *(ISSTA 2022)*. Association for Computing Machinery, New York, NY, USA, 519–531. doi:10.1145/3533767.3534371

[12] Zhaoyang Chu, Yao Wan, Qian Li, Yang Wu, Hongyu Zhang, Yulei Sui, Guandong Xu, and Hai Jin. 2024. Graph Neural Networks for Vulnerability Detection: A Counterfactual Explanation. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis* (Vienna, Austria) *(ISSTA 2024)*. Association for Computing Machinery, New York, NY, USA, 389–401. doi:10.1145/3650212.3652136

[13] Roland Croft, M. Ali Babar, and M. Mehdi Kholoosi. 2023. Data Quality for Software Vulnerability Datasets. In *Proceedings of the 45th International Conference on Software Engineering* (Melbourne, Victoria, Australia) *(ICSE '23)*. IEEE Press, 121–133. doi:10.1109/ICSE48619.2023.00022

[14] Roland Croft, Yongzheng Xie, and Muhammad Ali Babar. 2023. Data Preparation for Software Vulnerability Prediction: A Systematic Literature Review. *IEEE Transactions on Software Engineering* 49, 3 (2023), 1044–1063. doi:10.1109/TSE.2022.3171202

[15] Lei Cui, Zhiyu Hao, Yang Jiao, Haiqiang Fei, and Xiaochun Yun. 2021. VulDetector: Detecting Vulnerabilities Using Weighted Feature Graph Comparison. *IEEE Transactions on Information Forensics and Security* 16 (2021), 2004–2017. doi:10.1109/TIFS.2020.3047756

[16] Hoa Khanh Dam, Truyen Tran, Trang Pham, Shien Wee Ng, John Grundy, and Aditya Ghose. 2021. Automatic Feature Learning for Predicting Vulnerable Software Components. *IEEE Transactions on Software Engineering* 47, 1 (2021), 67–85. doi:10.1109/TSE.2018.2881961

[17] Yangruibo Ding, Saikat Chakraborty, Luca Buratti, Saurabh Pujar, Alessandro Morari, Gail Kaiser, and Baishakhi Ray. 2023. CONCORD: Clone-Aware Contrastive Learning for Source Code. In *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis* (Seattle, WA, USA) *(ISSTA 2023)*. Association for Computing

Machinery, New York, NY, USA, 26–38. doi:10.1145/3597926.3598035

[18] Siyue Feng, Yueming Wu, Wenjie Xue, Sikui Pan, Deqing Zou, Yang Liu, and Hai Jin. 2024. FIRE: combining multi-stage filtering with taint analysis for scalable recurring vulnerability detection. In *Proceedings of the 33rd USENIX Conference on Security Symposium* (Philadelphia, PA, USA) *(SEC '24)*. USENIX Association, USA, Article 105, 18 pages.

[19] Michael Fu, Van Nguyen, Chakkrit Kla Tantithamthavorn, Trung Le, and Dinh Phung. 2023. VulExplainer: A Transformer-Based Hierarchical Distillation for Explaining Vulnerability Types. *IEEE Transactions on Software Engineering* 49, 10 (2023), 4550–4565. doi:10.1109/TSE.2023.3305244

[20] Michael Fu, Chakkrit Tantithamthavorn, Trung Le, Van Nguyen, and Dinh Phung. 2022. VulRepair: a T5-based automated software vulnerability repair. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Singapore, Singapore) *(ESEC/FSE 2022)*. Association for Computing Machinery, New York, NY, USA, 935–947. doi:10.1145/3540250.3549098

[21] Anastasiia Grishina, Max Hort, and Leon Moonen. 2023. The EarlyBIRD Catches the Bug: On Exploiting Early Layers of Encoder Models for More Efficient Code Classification. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (San Francisco, CA, USA) *(ESEC/FSE 2023)*. Association for Computing Machinery, New York, NY, USA, 895–907. doi:10.1145/3611643.3616304

[22] Binfa Gui, Wei Song, Hailong Xiong, and Jeff Huang. 2022. Automated Use-After-Free Detection and Exploit Mitigation: How Far Have We Gone? *IEEE Transactions on Software Engineering* 48, 11 (2022), 4569–4589. doi:10.1109/TSE.2021.3121994

[23] Soumyadeep Hore, Fariha Moomtaheen, Ankit Shah, and Xinming Ou. 2023. Towards Optimal Triage and Mitigation of Context-Sensitive Cyber Vulnerabilities. *IEEE Transactions on Dependable and Secure Computing* 20, 2 (2023), 1270–1285. doi:10.1109/TDSC.2022.3152164

[24] Soneya Binta Hossain, Nan Jiang, Qiang Zhou, Xiaopeng Li, Wen-Hao Chiang, Yingjun Lyu, Hoan Nguyen, and Omer Tripp. 2024. A Deep Dive into Large Language Models for Automated Bug Localization and Repair. *Proc. ACM Softw. Eng.* 1, FSE, Article 66 (July 2024), 23 pages. doi:10.1145/3660773

[25] 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* (Seattle, WA, USA) *(ISSTA 2023)*. Association for Computing Machinery, New York, NY, USA, 1407–1419. doi:10.1145/3597926.3598145

[26] Kaifeng Huang, Chenhao Lu, Yiheng Cao, Bihuan Chen, and Xin Peng. 2024. VMud: Detecting Recurring Vulnerabilities with Multiple Fixing Functions via Function Selection and Semantic Equivalent Statement Matching. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security* (Salt Lake City, UT, USA) *(CCS '24)*. Association for Computing Machinery, New York, NY, USA, 3958–3972. doi:10.1145/3658644.3690372

[27] Yuanming Huang, Mingshu He, Xiaojuan Wang, and Jie Zhang. 2024. HeVulD: A Static Vulnerability Detection Method Using Heterogeneous Graph Code Representation. *IEEE Transactions on Information Forensics and Security* 19 (2024), 9129–9144. doi:10.1109/TIFS.2024.3457162

[28] Yuan Jiang, Yujian Zhang, Xiaohong Su, Christoph Treude, and Tiantian Wang. 2024. StagedVulBERT: Multigranular Vulnerability Detection With a Novel Pretrained Code Model. *IEEE Transactions on Software Engineering* 50, 12 (2024), 3454–3471. doi:10.1109/TSE.2024.3493245

[29] Yi Li, Shaohua Wang, and Tien N. Nguyen. 2021. Vulnerability detection with fine-grained interpretations. In *Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Athens, Greece) *(ESEC/FSE 2021)*. Association for Computing Machinery, New York, NY, USA, 292–303. doi:10.1145/3468264.3468597

[30] Yi Li, Shaohua Wang, and Tien N. Nguyen. 2022. Fault localization to detect co-change fixing locations. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (Singapore, Singapore) *(ESEC/FSE 2022)*. Association for Computing Machinery, New York, NY, USA, 659–671. doi:10.1145/3540250.3549137

[31] Yi Li, Aashish Yadavally, Jiaxing Zhang, Shaohua Wang, and Tien N. Nguyen. 2023. Commit-Level, Neural Vulnerability Detection and Assessment. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (San Francisco, CA, USA) *(ESEC/FSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1024–1036. doi:10.1145/3611643.3616346

[32] Yao Li, Tao Zhang, Xiapu Luo, Haipeng Cai, Sen Fang, and Dawei Yuan. 2023. Do Pretrained Language Models Indeed Understand Software Engineering Tasks? *IEEE Transactions on Software Engineering* 49, 10 (2023), 4639–4655. doi:10.1109/TSE.2023.3308952

[33] Zhen Li, Ning Wang, Deqing Zou, Yating Li, Ruqian Zhang, Shouhuai Xu, Chao Zhang, and Hai Jin. 2024. On the Effectiveness of Function-Level Vulnerability Detectors for Inter-Procedural Vulnerabilities. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 157, 12 pages. doi:10.1145/3597503.3639218

[34] Zhen Li, Ruqian Zhang, Deqing Zou, Ning Wang, Yating Li, Shouhuai Xu, Chen Chen, and Hai Jin. 2023. Robin: A Novel Method to Produce Robust Interpreters for Deep Learning-Based Code Classifiers. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 27–39. doi:10.1109/ASE56229.2023.00164

[35] Zhen Li, Deqing Zou, Shouhuai Xu, Zhaoxuan Chen, Yawei Zhu, and Hai Jin. 2022. VulDeeLocator: A Deep Learning-Based Fine-Grained Vulnerability Detector. *IEEE Transactions on Dependable and Secure Computing* 19, 4 (2022), 2821–2837. doi:10.1109/TDSC.2021.3076142

[36] Zhen Li, Deqing Zou, Shouhuai Xu, Hai Jin, Yawei Zhu, and Zhaoxuan Chen. 2022. SySeVR: A Framework for Using Deep Learning to Detect Software Vulnerabilities. *IEEE Transactions on Dependable and Secure Computing* 19, 4 (2022), 2244–2258. doi:10.1109/TDSC.2021.3051525

[37] Chen Liang, Qiang Wei, Zirui Jiang, Yisen Wang, and Jiang Du. 2024. A Source Code Vulnerability Detection Method Based on Adaptive Graph Neural Networks. In *2024 39th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*. 187–196.

[38] Guanjun Lin, Jun Zhang, Wei Luo, Lei Pan, Olivier De Vel, Paul Montague, and Yang Xiang. 2021. Software Vulnerability Discovery via Learning Multi-Domain Knowledge Bases. *IEEE Transactions on Dependable and Secure Computing* 18, 5 (2021), 2469–2485. doi:10.1109/TDSC.2019.2954088

[39] Jiaxing Liu, Chaofeng Sha, and Xin Peng. 2023. An Empirical Study of Parameter-Efficient Fine-Tuning Methods for Pre-Trained Code Models. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 397–408. doi:10.1109/ASE56229.2023.00125

[40] Peiyu Liu, Junming Liu, Lirong Fu, Kangjie Lu, Yifan Xia, Xuhong Zhang, Wenzhi Chen, Haiqin Weng, Shouling Ji, and Wenhai Wang. 2024. Exploring ChatGPT's capabilities on vulnerability management. In *Proceedings of the 33rd USENIX Conference on Security Symposium* (Philadelphia, PA, USA) *(SEC '24)*. USENIX Association, USA, Article 46, 18 pages.

[41] Shangqing Liu. 2020. A Unified Framework to Learn Program Semantics with Graph Neural Networks. In *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 1364–1366.

[42] Shigang Liu, Di Cao, Junae Kim, Tamas Abraham, Paul Montague, Seyit Camtepe, Jun Zhang, and Yang Xiang. 2024. EaTVul: ChatGPT-based evasion attack against software vulnerability detection. In *Proceedings of the 33rd USENIX Conference on Security Symposium* (Philadelphia, PA, USA) *(SEC '24)*. USENIX Association, USA, Article 411, 18 pages.

[43] Shigang Liu, Guanjun Lin, Lizhen Qu, Jun Zhang, Olivier De Vel, Paul Montague, and Yang Xiang. 2022. CD-VulD: Cross-Domain Vulnerability Discovery Based on Deep Domain Adaptation. *IEEE Transactions on Dependable and Secure Computing* 19, 1 (2022), 438–451. doi:10.1109/TDSC.2020.2984505

[44] Yiming Liu, Cen Zhang, Feng Li, Yeting Li, Jianhua Zhou, Jian Wang, Lanlan Zhan, Yang Liu, and Wei Huo. 2024. Semantic-Enhanced Static Vulnerability Detection in Baseband Firmware. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 173, 12 pages. doi:10.1145/3597503.3639158

[45] Zhongxin Liu, Zhijie Tang, Junwei Zhang, Xin Xia, and Xiaohu Yang. 2024. Pre-training by Predicting Program Dependencies for Vulnerability Analysis Tasks. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 151, 13 pages. doi:10.1145/3597503.3639142

[46] Zhenhao Luo, Pengfei Wang, Baosheng Wang, Jack Tt, Wei Xie, Xu Zhou, Danjun Liu, and Kai Lu. 2023. VulHawk: Cross-architecture Vulnerability Detection with Entropy-based Binary Code Search. doi:10.14722/ndss.2023.24415

[47] Triet Huynh Minh Le, David Hin, Roland Croft, and M. Ali Babar. 2021. DeepCVA: Automated Commit-level Vulnerability Assessment with Deep Multi-task Learning. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 717–729. doi:10.1109/ASE51524.2021.9678622

[48] Yisroel Mirsky, George Macon, Michael Brown, Carter Yagemann, Matthew Pruett, Evan Downing, Sukarno Mertoguno, and Wenke Lee. 2023. VulChecker: graph-based vulnerability localization in source code. In *Proceedings of the 32nd USENIX Conference on Security Symposium* (Anaheim, CA, USA) *(SEC '23)*. USENIX Association, USA, Article 367, 18 pages.

[49] Yazdan Movahedi, Michel Cukier, and Ilir Gashi. 2022. Predicting the Discovery Pattern of Publically Known Exploited Vulnerabilities. *IEEE Transactions on Dependable and Secure Computing* 19, 2 (2022), 1181–1193. doi:10.1109/TDSC.2020.3014872

[50] Van Nguyen, Trung Le, Chakkrit Tantithamthavorn, John Grundy, and Dinh Phung. 2024. Deep Domain Adaptation With Max-Margin Principle for Cross-Project Imbalanced Software Vulnerability Detection. *ACM Trans. Softw. Eng. Methodol.* 33, 6, Article 162 (June 2024), 34 pages. doi:10.1145/3664602

[51] Chao Ni, Xinrong Guo, Yan Zhu, Xiaodan Xu, and Xiaohu Yang. 2023. Function-Level Vulnerability Detection Through Fusing Multi-Modal Knowledge. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 1911–1918. doi:10.1109/ASE56229.2023.00084

[52] 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 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (San Francisco, CA, USA) *(ESEC/FSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1611–1622. doi:10.1145/3611643.3616358

[53] Xu Nie, Ningke Li, Kailong Wang, Shangguang Wang, Xiapu Luo, and Haoyu Wang. 2023. Understanding and Tackling Label Errors in Deep Learning-Based Vulnerability Detection (Experience Paper). In *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis* (Seattle, WA, USA) *(ISSTA 2023)*. Association for Computing Machinery, New York, NY, USA, 52–63. doi:10.1145/3597926.3598037

[54] Yu Nong, Richard Fang, Guangbei Yi, Kunsong Zhao, Xiapu Luo, Feng Chen, and Haipeng Cai. 2024. VGX: Large-Scale Sample Generation for Boosting Learning-Based Software Vulnerability Analyses. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 149, 13 pages. doi:10.1145/3597503.3639116

[55] Yu Nong, Rainy Sharma, Abdelwahab Hamou-Lhadj, Xiapu Luo, and Haipeng Cai. 2023. Open Science in Software Engineering: A Study on Deep Learning-Based Vulnerability Detection. *IEEE Transactions on Software Engineering* 49, 4 (2023), 1983–2005. doi:10.1109/TSE.2022.3207149

[56] Fangcheng Qiu, Zhongxin Liu, Xing Hu, Xin Xia, Gang Chen, and Xinyu Wang. 2024. Vulnerability Detection via Multiple-Graph-Based Code Representation. *IEEE Transactions on Software Engineering* 50, 8 (2024), 2178–2199. doi:10.1109/TSE.2024.3427815

[57] Yuxin Qiu, Jie Hu, Qian Zhang, and Heng Yin. 2024. Calico: Automated Knowledge Calibration and Diagnosis for Elevating AI Mastery in Code Tasks. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis* (Vienna, Austria) *(ISSTA 2024)*. Association for Computing Machinery, New York, NY, USA, 1785–1797. doi:10.1145/3650212.3680399

[58] Md Mahbubur Rahman, Ira Ceka, Chengzhi Mao, Saikat Chakraborty, Baishakhi Ray, and Wei Le. 2024. Towards Causal Deep Learning for Vulnerability Detection. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 153, 11 pages. doi:10.1145/3597503.3639170

[59] Cedric Richter and Heike Wehrheim. 2023. How to Train Your Neural Bug Detector: Artificial vs Real Bugs. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 1036–1048. doi:10.1109/ASE56229.2023.00104

[60] Niklas Risse. 2023. Detecting Overfitting of Machine Learning Techniques for Automatic Vulnerability Detection. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (San Francisco, CA, USA) *(ESEC/FSE 2023)*. Association for Computing Machinery, New York, NY, USA, 2189–2191. doi:10.1145/3611643.3617845

[61] Niklas Risse and Marcel Böhme. 2024. Uncovering the limits of machine learning for automatic vulnerability detection. In *Proceedings of the 33rd USENIX Conference on Security Symposium* (Philadelphia, PA, USA) *(SEC '24)*. USENIX Association, USA, Article 238, 18 pages.

[62] Adriana Sejfia, Satyaki Das, Saad Shafiq, and Nenad Medvidović. 2024. Toward Improved Deep Learning-based Vulnerability Detection. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 62, 12 pages. doi:10.1145/3597503.3608141

[63] Changjie Shao, Gaolei Li, Jun Wu, and Xi Zheng. 2024. Exploring Semantic Redundancy using Backdoor Triggers: A Complementary Insight into the Challenges Facing DNN-based Software Vulnerability Detection. *ACM Trans. Softw. Eng. Methodol.* 33, 4, Article 92 (April 2024), 28 pages. doi:10.1145/3640333

[64] Miaomiao Shao and Yuxin Ding. 2024. FVD-DPM: fine-grained vulnerability detection via conditional diffusion probabilistic models. In *Proceedings of the 33rd USENIX Conference on Security Symposium* (Philadelphia, PA, USA) *(SEC '24)*. USENIX Association, USA, Article 412, 18 pages.

[65] Samiha Shimmi, Ashiqur Rahman, Mohan Gadde, Hamed Okhravi, and Mona Rahimi. 2024. VulSim: leveraging similarity of multi-dimensional neighbor embeddings for vulnerability detection. In *Proceedings of the 33rd USENIX Conference on Security Symposium* (Philadelphia, PA, USA) *(SEC '24)*. USENIX Association, USA, Article 100, 18 pages.

[66] Benjamin Steenhoek, Hongyang Gao, and Wei Le. 2024. Dataflow Analysis-Inspired Deep Learning for Efficient Vulnerability Detection. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 16, 13 pages. doi:10.1145/3597503.3623345

[67] Benjamin Steenhoek, Md Mahbubur Rahman, Richard Jiles, and Wei Le. 2023. An Empirical Study of Deep Learning Models for Vulnerability Detection. In *Proceedings of the 45th International Conference on Software Engineering* (Melbourne, Victoria, Australia) *(ICSE '23)*. IEEE Press, 2237–2248. doi:10.1109/ICSE48619.2023.00188

[68] Sahil Suneja, Yufan Zhuang, Yunhui Zheng, Jim Laredo, Alessandro Morari, and Udayan Khurana. 2023. Incorporating Signal Awareness in Source Code Modeling: An Application to Vulnerability Detection. *ACM Trans. Softw. Eng. Methodol.* 32, 6, Article 145 (Sept. 2023), 40 pages. doi:10.1145/3597202

[69] Saad Ullah, Mingji Han, Saurabh Pujar, Hammond Pearce, Ayse Coskun, and Gianluca Stringhini. 2024. LLMs Cannot Reliably Identify and Reason About Security Vulnerabilities (Yet?): A Comprehensive Evaluation, Framework, and Benchmarks. In *2024 IEEE Symposium on Security and Privacy (SP)*. 862–880. doi:10.1109/SP54263.2024.00210

[70] Huanting Wang, Zhanyong Tang, Shin Hwei Tan, Jie Wang, Yuzhe Liu, Hejun Fang, Chunwei Xia, and Zheng Wang. 2024. Combining Structured Static Code Information and Dynamic Symbolic Traces for Software Vulnerability Prediction. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 169, 13 pages. doi:10.1145/3597503.3639212

[71] Huanting Wang, Guixin Ye, Zhanyong Tang, Shin Hwei Tan, Songfang Huang, Dingyi Fang, Yansong Feng, Lizhong Bian, and Zheng Wang. 2021. Combining Graph-Based Learning With Automated Data Collection for Code Vulnerability Detection. *IEEE Transactions on Information Forensics and Security* 16 (2021), 1943–1958. doi:10.1109/TIFS.2020.3044773

[72] Wentao Wang, Faryn Dumont, Nan Niu, and Glen Horton. 2022. Detecting Software Security Vulnerabilities Via Requirements Dependency Analysis. *IEEE Transactions on Software Engineering* 48, 5 (2022), 1665–1675. doi:10.1109/TSE.2020.3030745

[73] Wenbo Wang, Tien N. Nguyen, Shaohua Wang, Yi Li, Jiyuan Zhang, and Aashish Yadavally. 2023. DeepVD: Toward Class-Separation Features for Neural Network Vulnerability Detection. In *Proceedings of the 45th International Conference on Software Engineering* (Melbourne, Victoria, Australia) *(ICSE '23)*. IEEE Press, 2249–2261. doi:10.1109/ICSE48619.2023.00189

[74] Xin-Cheng Wen, Yupan Chen, Cuiyun Gao, Hongyu Zhang, Jie M. Zhang, and Qing Liao. 2023. Vulnerability Detection with Graph Simplification and Enhanced Graph Representation Learning. In *Proceedings of the 45th International Conference on Software Engineering* (Melbourne, Victoria, Australia) *(ICSE '23)*. IEEE Press, 2275–2286. doi:10.1109/ICSE48619.2023.00191

[75] Xin-Cheng Wen, Cuiyun Gao, Shuzheng Gao, Yang Xiao, and Michael R. Lyu. 2024. SCALE: Constructing Structured Natural Language Comment Trees for Software Vulnerability Detection. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis* (Vienna, Austria) *(ISSTA 2024)*. Association for Computing Machinery, New York, NY, USA, 235–247. doi:10.1145/3650212.3652124

[76] Xin-Cheng Wen, Cuiyun Gao, Feng Luo, Haoyu Wang, Ge Li, and Qing Liao. 2024. LIVABLE: Exploring Long-Tailed Classification of Software Vulnerability Types. *IEEE Transactions on Software Engineering* 50, 6 (2024), 1325–1339. doi:10.1109/TSE.2024.3382361

[77] Xin-Cheng Wen, Cuiyun Gao, Jiaxin Ye, Yichen Li, Zhihong Tian, Yan Jia, and Xuan Wang. 2024. Meta-Path Based Attentional Graph Learning Model for Vulnerability Detection. *IEEE Transactions on Software Engineering* 50, 3 (2024), 360–375. doi:10.1109/TSE.2023.3340267

[78] Xin-Cheng Wen, Xinchen Wang, Cuiyun Gao, Shaohua Wang, Yang Liu, and Zhaoquan Gu. 2023. When Less is Enough: Positive and Unlabeled Learning Model for Vulnerability Detection. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 345–357. doi:10.1109/ASE56229.2023.00144

[79] Seunghoon Woo, Eunjin Choi, Heejo Lee, and Hakjoo Oh. 2023. V1SCAN: discovering 1-day vulnerabilities in reused C/C++ open-source software components using code classification techniques. In *Proceedings of the 32nd USENIX Conference on Security Symposium* (Anaheim, CA, USA) *(SEC '23)*. USENIX Association, USA, Article 366, 16 pages.

[80] Seunghoon Woo, Hyunji Hong, Eunjin Choi, and Heejo Lee. 2022. MOVERY: A Precise Approach for Modified Vulnerable Code Clone Discovery from Modified Open-Source Software Components. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 3037–3053. https://www.usenix.org/conference/usenixsecurity22/presentation/woo

[81] Bozhi Wu, Shangqing Liu, Yang Xiao, Zhiming Li, Jun Sun, and Shang-Wei Lin. 2023. Learning Program Semantics for Vulnerability Detection via Vulnerability-Specific Inter-procedural Slicing. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (San Francisco, CA, USA) *(ESEC/FSE 2023)*. Association for Computing Machinery, New York, NY, USA, 1371–1383. doi:10.1145/3611643.3616351

[82] Tongshuai Wu, Liwei Chen, Gewangzi Du, Dan Meng, and Gang Shi. 2024. UltraVCS: Ultra-Fine-Grained Variable-Based Code Slicing for Automated Vulnerability Detection. *IEEE Transactions on Information Forensics and Security* 19 (2024), 3986–4000. doi:10.1109/TIFS.2024.3374219

[83] Yulun Wu, Ming Wen, Zeliang Yu, Xiaochen Guo, and Hai Jin. 2024. Effective Vulnerable Function Identification based on CVE Description Empowered by Large Language Models. In *2024 39th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 393–405.

[84] Yueming Wu, Deqing Zou, Shihan Dou, Wei Yang, Duo Xu, and Hai Jin. 2022. VulCNN: an image-inspired scalable vulnerability detection system. In *Proceedings of the 44th International Conference on Software Engineering* (Pittsburgh, Pennsylvania) *(ICSE '22)*. Association for Computing Machinery, New York, NY, USA, 2365–2376. doi:10.1145/3510003.3510229

[85] Peng Xiao, Qibin Xiao, Xusheng Zhang, Yumei Wu, and Fengyu Yang. 2024. Vulnerability Detection Based on Enhanced Graph Representation Learning. *IEEE Transactions on Information Forensics and Security* 19 (2024), 5120–5135. doi:10.1109/TIFS.2024.3392536

[86] Yang Xiao, Bihuan Chen, Chendong Yu, Zhengzi Xu, Zimu Yuan, Feng Li, Binghong Liu, Yang Liu, Wei Huo, Wei Zou, and Wenchang Shi. 2020. MVP: detecting vulnerabilities using patch-enhanced vulnerability signatures. In *Proceedings of the 29th USENIX Conference on Security Symposium (SEC'20)*. USENIX Association, USA, Article 66, 18 pages.

[87] Xu Yang, Shaowei Wang, Yi Li, and Shaohua Wang. 2023. Does Data Sampling Improve Deep Learning-Based Vulnerability Detection? Yeas! and Nays!. In *Proceedings of the 45th International Conference on Software Engineering* (Melbourne, Victoria, Australia) *(ICSE '23)*. IEEE Press, 2287–2298. doi:10.1109/ICSE48619.2023.00192

[88] Yulong Yang, Haoran Fan, Chenhao Lin, Qian Li, Zhengyu Zhao, and Chao Shen. 2024. Exploiting the Adversarial Example Vulnerability of Transfer Learning of Source Code. *IEEE Transactions on Information Forensics and Security* 19 (2024), 5880–5894. doi:10.1109/TIFS.2024.3402153

[89] Yanming Yang, Xing Hu, Zhipeng Gao, Jinfu Chen, Chao Ni, Xin Xia, and David Lo. 2024. Federated Learning for Software Engineering: A Case Study of Code Clone Detection and Defect Prediction. *IEEE Transactions on Software Engineering* 50, 2 (2024), 296–321. doi:10.1109/TSE.2023.3347898

[90] Xin Yin, Chao Ni, and Shaohua Wang. 2024. Multitask-Based Evaluation of Open-Source LLM on Software Vulnerability. *IEEE Transactions on Software Engineering* 50, 11 (2024), 3071–3087. doi:10.1109/TSE.2024.3470333

[91] Zijing Yin, Yiwen Xu, Fuchen Ma, Haohao Gao, Lei Qiao, and Yu Jiang. 2023. Scanner++: Enhanced Vulnerability Detection of Web Applications with Attack Intent Synchronization. *ACM Trans. Softw. Eng. Methodol.* 32, 1, Article 7 (Feb. 2023), 30 pages. doi:10.1145/3517036

[92] Jianjia Yu, Song Li, Junmin Zhu, and Yinzhi Cao. 2023. CoCo: Efficient Browser Extension Vulnerability Detection via Coverage-guided, Concurrent Abstract Interpretation. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 2441–2455. doi:10.1145/3576915.3616584

[93] Xiao Yu, Guancheng Lin, Xing Hu, Jacky Wai Keung, and Xin Xia. 2024. Less is More: Unlocking Semi-Supervised Deep Learning for Vulnerability Detection. *ACM Trans. Softw. Eng. Methodol.* (Oct. 2024). doi:10.1145/3699602 Just Accepted.

[94] Xiao Yu, Lei Liu, Xing Hu, Jacky Wai Keung, Jin Liu, and Xin Xia. 2024. Fight Fire With Fire: How Much Can We Trust ChatGPT on Source Code-Related Tasks? *IEEE Transactions on Software Engineering* 50, 12 (2024), 3435–3453. doi:10.1109/TSE.2024.3492204

[95] Zhe Yu, Christopher Theisen, Laurie Williams, and Tim Menzies. 2021. Improving Vulnerability Inspection Efficiency Using Active Learning. *IEEE Transactions on Software Engineering* 47, 11 (2021), 2401–2420. doi:10.1109/TSE.2019.2949275

[96] Bin Yuan, Yifan Lu, Yilin Fang, Yueming Wu, Deqing Zou, Zhen Li, Zhi Li, and Hai Jin. 2023. Enhancing Deep Learning-Based Vulnerability Detection by Building Behavior Graph Model. In *Proceedings of the 45th International Conference on Software Engineering* (Melbourne, Victoria, Australia) *(ICSE '23)*. IEEE Press, 2262–2274. doi:10.1109/ICSE48619.2023.00190

[97] Chunyong Zhang, Bin Liu, Yang Xin, and Liangwei Yao. 2023. CPVD: Cross Project Vulnerability Detection Based on Graph Attention Network and Domain Adaptation. *IEEE Transactions on Software Engineering* 49, 8 (2023), 4152–4168. doi:10.1109/TSE.2023.3285910

[98] Huangzhao Zhang, Zhiyi Fu, Ge Li, Lei Ma, Zhehao Zhao, Hua'an Yang, Yizhe Sun, Yang Liu, and Zhi Jin. 2022. Towards Robustness of Deep Program Processing Models—Detection, Estimation, and Enhancement. *ACM Trans. Softw. Eng. Methodol.* 31, 3, Article 50 (April 2022), 40 pages. doi:10.1145/3511887

[99] Jian Zhang, Shangqing Liu, Xu Wang, Tianlin Li, and Yang Liu. 2023. Learning to Locate and Describe Vulnerabilities. In *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 332–344. doi:10.1109/ASE56229.2023.00045

[100] Junwei Zhang, Zhongxin Liu, Xing Hu, Xin Xia, and Shanping Li. 2023. Vulnerability Detection by Learning From Syntax-Based Execution Paths of Code. *IEEE Transactions on Software Engineering* 49, 8 (2023), 4196–4212. doi:10.1109/TSE.2023.3286586

[101] Jian Zhang, Xu Wang, Hongyu Zhang, Hailong Sun, Xudong Liu, Chunming Hu, and Yang Liu. 2023. Detecting Condition-Related Bugs with Control Flow Graph Neural Network. In *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis* (Seattle, WA, USA) *(ISSTA 2023)*. Association for Computing

Machinery, New York, NY, USA, 1370–1382. doi:10.1145/3597926.3598142

[102] Tiehua Zhang, Rui Xu, Jianping Zhang, Yuze Liu, Xin Chen, Jun Yin, and Xi Zheng. 2024. DSHGT: Dual-Supervisors Heterogeneous Graph Transformer—A Pioneer Study of Using Heterogeneous Graph Learning for Detecting Software Vulnerabilities. *ACM Trans. Softw. Eng. Methodol.* 33, 8, Article 202 (Nov. 2024), 31 pages. doi:10.1145/3674729

[103] Lida Zhao, Sen Chen, Zhengzi Xu, Chengwei Liu, Lyuye Zhang, Jiahui Wu, Jun Sun, and Yang Liu. 2023. Software Composition Analysis for Vulnerability Detection: An Empirical Study on Java Projects. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (San Francisco, CA, USA) *(ESEC/FSE 2023)*. Association for Computing Machinery, New York, NY, USA, 960–972. doi:10.1145/3611643.3616299

[104] Yu Zhao, Lina Gong, Zhiqiu Huang, Yongwei Wang, Mingqiang Wei, and Fei Wu. 2024. Coding-PTMs: How to Find Optimal Code Pre-trained Models for Code Embedding in Vulnerability Detection?. In *2024 39th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. 1732–1744.

[105] Xin Zhou, Sicong Cao, Xiaobing Sun, and David Lo. 2024. Large Language Model for Vulnerability Detection and Repair: Literature Review and the Road Ahead. *ACM Trans. Softw. Eng. Methodol.* (Dec. 2024). doi:10.1145/3708522 Just Accepted.

[106] Xin Zhou, Kisub Kim, Bowen Xu, Donggyun Han, and David Lo. 2024. Out of Sight, Out of Mind: Better Automatic Vulnerability Repair by Broadening Input Ranges and Sources. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (Lisbon, Portugal) *(ICSE '24)*. Association for Computing Machinery, New York, NY, USA, Article 88, 13 pages. doi:10.1145/3597503.3639222

[107] Deqing Zou, Yutao Hu, Wenke Li, Yueming Wu, Haojun Zhao, and Hai Jin. 2022. mVulPreter: A Multi-Granularity Vulnerability Detection System With Interpretations. *IEEE Transactions on Dependable and Secure Computing* (2022), 1–12. doi:10.1109/TDSC.2022.3199769

[108] Deqing Zou, Sujuan Wang, Shouhuai Xu, Zhen Li, and Hai Jin. 2021. μVulDeePecker: A Deep Learning-Based System for Multiclass Vulnerability Detection. *IEEE Transactions on Dependable and Secure Computing* 18, 5 (2021), 2224–2236. doi:10.1109/TDSC.2019.2942930

[109] 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, Article 23 (March 2021), 31 pages. doi:10.1145/3429444

Received 2024-10-29; accepted 2025-03-31