Appendix A: Primary Studies

Table 1: List of Primary Studies

#	Reference
[S1]	S. Neuhaus, T. Zimmermann, C. Holler, and A. Zeller, “Predicting vulnerable software components,” in Proceedings of the 14th ACM conference on Computer and communications security, 2007, pp. 529–540. doi: https://doi.org/10.1145/1315245.1315311.
[S2]	Z. Li, D. Zou, S. Xu, Z. Chen, Y. Zhu, and H. Jin, “Vuldeelocator: a deep learning-based fine-grained vulnerability detector,” IEEE Transactions on Dependable and Secure Computing, 2021, doi: https://doi.org/10.1109/TDSC.2021.3076142.
[S3]	J. D. Pereira, J. R. Campos, and M. Vieira, “Machine learning to combine static analysis alerts with software metrics to detect security vulnerabilities: An empirical study,” in 2021 17th European Dependable Computing Conference (EDCC), 2021, pp. 1–8. doi: https://doi.org/10.1109/EDCC53658.2021.00008.
[S4]	W. Xiaomeng, Z. Tao, W. Runpu, X. Wei, and H. Changyu, “CPGVA: code property graph based vulnerability analysis by deep learning,” in 2018 10th International Conference on Advanced Infocomm Technology (ICAIT), 2018, pp. 184–188. doi: https://doi.org/10.1109/ICAIT.2018.8686548.
[S5]	X. Xia, Y. Wang, and Y. Yang, “Source Code Vulnerability Detection Based On SAR-GIN,” in 2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT), 2021, pp. 1144–1149. doi: https://doi.org/10.1109/CECIT53797.2021.00202.
[S6]	W. Zheng, A. O. A. Semasaba, X. Wu, S. A. Agyemang, T. Liu, and Y. Ge, “Representation vs. Model: What Matters Most for Source Code Vulnerability Detection,” in 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 647–653. doi: https://doi.org/10.1109/SANER50967.2021.00082.
[S7]	L. Wartschinski, Y. Noller, T. Vogel, T. Kehrer, and L. Grunske, “VUDENC: Vulnerability Detection with Deep Learning on a Natural Codebase for Python,” Information and Software Technology, vol. 144, p. 106809, 2022, doi: https://doi.org/10.1016/j.infsof.2021.106809.
[S8]	J. Iqbal, T. Firdous, A. K. Shrivastava, and I. Saraf, “Modelling and predicting software vulnerabilities using a sigmoid function,” International Journal of Information Technology, vol. 14, no. 2, pp. 649–655, 2022, doi: https://doi.org/10.1007/s41870-021-00844-2.
[S9]	S. S. Murtaza, W. Khreich, A. Hamou-Lhadj, and A. B. Bener, “Mining trends and patterns of software vulnerabilities,” Journal of Systems and Software, vol. 117, pp. 218–228, 2016, doi: https://doi.org/10.1016/j.jss.2016.02.048.
[S10]	A. Bagheri and P. Hegedűs, “A comparison of different source code representation methods for vulnerability prediction in Python,” in International Conference on the Quality of Information and Communications Technology, 2021, pp. 267–281. doi: https://doi.org/10.1007/978-3-030-85347-1_20.
[S11]	A. Chatzipoulidis, D. Michalopoulos, and I. Mavridis, “Information infrastructure risk prediction through platform vulnerability analysis,” Journal of Systems and Software, vol. 106, pp. 28–41, 2015, doi: https://doi.org/10.1016/j.jss.2015.04.062.

[S12]	H. Yan, S. Luo, L. Pan, and Y. Zhang, “HAN-BSVD: a hierarchical attention network for binary software vulnerability detection,” Computers & Security, vol. 108, p. 102286, 2021, doi: https://doi.org/10.1016/j.cose.2021.102286.
[S13]	H. Perl et al., “Vccfinder: Finding potential vulnerabilities in open-source projects to assist code audits,” in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 426–437. doi: https://doi.org/10.1145/2810103.2813604.
[S14]	A. Mazuera-Rozo, A. Mojica-Hanke, M. Linares-Vásquez, and G. Bavota, “Shallow or deep? An empirical study on detecting vulnerabilities using deep learning,” in 2021 IEEE/ACM 29th International Conference on Program Comprehension (ICPC), 2021, pp. 276–287. doi: https://doi.org/10.1109/ICPC52881.2021.00034.
[S15]	N. Guo, X. Li, H. Yin, and Y. Gao, “Vulhunter: An automated vulnerability detection system based on deep learning and bytecode,” in International Conference on Information and Communications Security, 2019, pp. 199–218. doi: https://doi.org/10.1007/978-3-030-41579-2_12.
[S16]	K. Z. Sultana, A. Deo, and B. J. Williams, “Correlation analysis among java nano-patterns and software vulnerabilities,” in 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), 2017, pp. 69–76. doi: https://doi.org/10.1109/HASE.2017.18.
[S17]	A. K. Shrivastava, R. Sharma, and P. K. Kapur, “Vulnerability discovery model for a software system using stochastic differential equation,” in 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015, pp. 199–205. doi: 10.1109/ABLAZE.2015.7154992.
[S18]	A. Gkortzis, D. Mitropoulos, and D. Spinellis, “VulinOSS: A Dataset of Security Vulnerabilities in Open-Source Systems,” in Proceedings of the 15th International Conference on Mining Software Repositories, 2018, pp. 18–21. doi: 10.1145/3196398.3196454.
[S19]	L. K. Shar and H. B. K. Tan, “Predicting Common Web Application Vulnerabilities from Input Validation and Sanitization Code Patterns,” in Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering, 2012, pp. 310–313. doi: 10.1145/2351676.2351733.
[S20]	C. Theisen and L. Williams, “Better together: Comparing vulnerability prediction models,” Information and Software Technology, vol. 119, p. 106204, 2020, doi: https://doi.org/10.1016/j.infsof.2019.106204.
[S21]	S. Jeon and H. K. Kim, “AutoVAS: An automated vulnerability analysis system with a deep learning approach,” Computers & Security, vol. 106, p. 102308, 2021, doi: https://doi.org/10.1016/j.cose.2021.102308.
[S22]	C. Batur Şahin and L. Abualigah, “A novel deep learning-based feature selection model for improving the static analysis of vulnerability detection,” Neural Computing and Applications, vol. 33, no. 20, pp. 14049–14067, 2021, doi: https://doi.org/10.1007/s00521-021-06047-x.
[S23]	T. Nguyen et al., “Deep cost-sensitive kernel machine for binary software vulnerability detection,” in Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2020, pp. 164–177. doi: https://doi.org/10.1007/978-3-030-47436-2_13.
[S24]	S. Liu, G. Lin, Q.-L. Han, S. Wen, J. Zhang, and Y. Xiang, “DeepBalance: Deep-Learning and Fuzzy Oversampling for Vulnerability Detection,” IEEE Transactions on Fuzzy Systems, vol. 28, no. 7, pp. 1329–1343, 2020, doi: 10.1109/TFUZZ.2019.2958558.
[S25]	E. Yasasin, J. Prester, G. Wagner, and G. Schryen, “Forecasting IT security vulnerabilities – An empirical analysis,” Computers & Security, vol. 88, p. 101610, 2020, doi: https://doi.org/10.1016/j.cose.2019.101610.

[S26]	G. Lin, W. Xiao, L. Y. Zhang, S. Gao, Y. Tai, and J. Zhang, “Deep neural-based vulnerability discovery demystified: data, model and performance,” Neural Computing and Applications, vol. 33, no. 20, pp. 13287–13300, 2021, doi: https://doi.org/10.1007/s00521-021-05954-3.
[S27]	S.-W. Woo, H. Joh, O. H. Alhazmi, and Y. K. Malaiya, “Modeling vulnerability discovery process in Apache and IIS HTTP servers,” Computers & Security, vol. 30, no. 1, pp. 50–62, 2011, doi: https://doi.org/10.1016/j.cose.2010.10.007.
[S28]	Y. Movahedi, M. Cukier, and I. Gashi, “Vulnerability prediction capability: A comparison between vulnerability discovery models and neural network models,” Computers & Security, vol. 87, p. 101596, 2019, doi: https://doi.org/10.1016/j.cose.2019.101596.
[S29]	O. Alhazmi, Y. Malaiya, and I. Ray, “Security vulnerabilities in software systems: A quantitative perspective,” in IFIP Annual Conference on Data and Applications Security and Privacy, 2005, pp. 281–294. doi: https://doi.org/10.1007/11535706_21.
[S30]	Y. Zheng, K. Cheng, Z. Li, S. Pan, H. Zhu, and L. Sun, “A lightweight method for accelerating discovery of taint-style vulnerabilities in embedded systems,” in International Conference on Information and Communications Security, 2016, pp. 27–36. doi: https://doi.org/10.1007/978-3-319-50011-9_3.
[S31]	Y. Kansal, P. K. Kapur, U. Kumar, and D. Kumar, “Effort and coverage dependent vulnerability discovery modeling,” in 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), 2017, pp. 1–6. doi: 10.1109/TEL-NET.2017.8343550.
[S32]	Y. Shin and L. Williams, “An Empirical Model to Predict Security Vulnerabilities Using Code Complexity Metrics,” in Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, 2008, pp. 315–317. doi: 10.1145/1414004.1414065.
[S33]	H. Alves, B. Fonseca, and N. Antunes, “Experimenting Machine Learning Techniques to Predict Vulnerabilities,” in 2016 Seventh Latin-American Symposium on Dependable Computing (LADC), 2016, pp. 151–156. doi: 10.1109/LADC.2016.32.
[S34]	K. Gencer and F. Başçiftçi, “Time series forecast modeling of vulnerabilities in the android operating system using ARIMA and deep learning methods,” Sustainable Computing: Informatics and Systems, vol. 30, p. 100515, 2021, doi: https://doi.org/10.1016/j.suscom.2021.100515.
[S35]	A. Ozment, “Software security growth modeling: Examining vulnerabilities with reliability growth models,” in Quality of Protection, Springer, 2006, pp. 25–36. doi: https://doi.org/10.1007/978-0-387-36584-8_3.
[S36]	W. Lin and S. Cai, “An Empirical Study on Vulnerability Detection for Source Code Software based on Deep Learning,” in 2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2021, pp. 1159–1160. doi: 10.1109/QRS-C55045.2021.00173.
[S37]	D. Zou, Y. Zhu, S. Xu, Z. Li, H. Jin, and H. Ye, “Interpreting Deep Learning-Based Vulnerability Detector Predictions Based on Heuristic Searching,” ACM Trans. Softw. Eng. Methodol., vol. 30, no. 2, Mar. 2021, doi: 10.1145/3429444.
[S38]	M. N. Khalid, H. Farooq, M. Iqbal, M. T. Alam, and K. Rasheed, “Predicting web vulnerabilities in web applications based on machine learning,” in International Conference on Intelligent Technologies and Applications, 2018, pp. 473–484. doi: https://doi.org/10.1007/978-981-13-6052-7_41.
[S39]	T. Wu, L. Chen, G. Du, C. Zhu, and G. Shi, “Self-Attention based Automated Vulnerability Detection with Effective Data Representation,” in 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable

	Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2021, pp. 892–899. doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00126.
[S40]	R. Ferenc, P. Hegedűs, P. Gyimesi, G. Antal, D. Bán, and T. Gyimóthy, “Challenging Machine Learning Algorithms in Predicting Vulnerable JavaScript Functions,” in 2019 IEEE/ACM 7th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE), 2019, pp. 8–14. doi: 10.1109/RAISE.2019.00010.
[S41]	M. Zagane, M. K. Abdi, and M. Alenezi, “Deep Learning for Software Vulnerabilities Detection Using Code Metrics,” IEEE Access, vol. 8, pp. 74562–74570, 2020, doi: 10.1109/ACCESS.2020.2988557.
[S42]	S. Salimi, M. Ebrahimzadeh, and M. Kharrazi, “Improving Real-World Vulnerability Characterization with Vulnerable Slices,” in Proceedings of the 16th ACM International Conference on Predictive Models and Data Analytics in Software Engineering, 2020, pp. 11–20. doi: 10.1145/3416508.3417120.
[S43]	G. Bhandari, A. Naseer, and L. Moonen, “CVEfixes: Automated Collection of Vulnerabilities and Their Fixes from Open-Source Software,” in Proceedings of the 17th International Conference on Predictive Models and Data Analytics in Software Engineering, 2021, pp. 30–39. doi: 10.1145/3475960.3475985.
[S44]	G. Tang et al., “A Comparative Study of Neural Network Techniques for Automatic Software Vulnerability Detection,” in 2020 International Symposium on Theoretical Aspects of Software Engineering (TASE), 2020, pp. 1–8. doi: 10.1109/TASE49443.2020.00010.
[S45]	Z. Li, D. Zou, J. Tang, Z. Zhang, M. Sun, and H. Jin, “A Comparative Study of Deep Learning-Based Vulnerability Detection System,” IEEE Access, vol. 7, pp. 103184–103197, 2019, doi: 10.1109/ACCESS.2019.2930578.
[S46]	Y. Shin and L. Williams, “An Initial Study on the Use of Execution Complexity Metrics as Indicators of Software Vulnerabilities,” in Proceedings of the 7th International Workshop on Software Engineering for Secure Systems, 2011, pp. 1–7. doi: 10.1145/1988630.1988632.
[S47]	O. H. Alhazmi, Y. K. Malaiya, and I. Ray, “Measuring, analyzing and predicting security vulnerabilities in software systems,” Computers & Security, vol. 26, no. 3, pp. 219–228, 2007, doi: https://doi.org/10.1016/j.cose.2006.10.002.
[S48]	X. Cheng, H. Wang, J. Hua, G. Xu, and Y. Sui, “DeepWukong: Statically Detecting Software Vulnerabilities Using Deep Graph Neural Network,” ACM Trans. Softw. Eng. Methodol., vol. 30, no. 3, Apr. 2021, doi: 10.1145/3436877.
[S49]	M. A. Williams, R. C. Barranco, S. M. Naim, S. Dey, M. Shahriar Hossain, and M. Akbar, “A vulnerability analysis and prediction framework,” Computers & Security, vol. 92, p. 101751, 2020, doi: https://doi.org/10.1016/j.cose.2020.101751.
[S50]	Z. Zheng, B. Zhang, Y. Liu, J. Ren, X. Zhao, and Q. Wang, “An approach for predicting multiple-type overflow vulnerabilities based on combination features and a time series neural network algorithm,” Computers & Security, vol. 114, p. 102572, 2022, doi: https://doi.org/10.1016/j.cose.2021.102572.
[S51]	G. Lin et al., “Software Vulnerability Discovery via Learning Multi-Domain Knowledge Bases,” IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 5, pp. 2469–2485, 2021, doi: 10.1109/TDSC.2019.2954088.
[S52]	R. Li, C. Feng, X. Zhang, and C. Tang, “A Lightweight Assisted Vulnerability Discovery Method Using Deep Neural Networks,” IEEE Access, vol. 7, pp. 80079–80092, 2019, doi: 10.1109/ACCESS.2019.2923227.

[S53]	D. Last, “Using historical software vulnerability data to forecast future vulnerabilities,” in 2015 Resilience Week (RWS), 2015, pp. 1–7. doi: 10.1109/RWEEK.2015.7287429.
[S54]	J. Kim, D. Hubczenko, and P. Montague, “Towards attention based vulnerability discovery using source code representation,” in International Conference on Artificial Neural Networks, 2019, pp. 731–746. doi: https://doi.org/10.1007/978-3-030-30490-4_58.
[S55]	D. Meng et al., “Bran: Reduce Vulnerability Search Space in Large Open Source Repositories by Learning Bug Symptoms,” in Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, 2021, pp. 731–743. doi: 10.1145/3433210.3453115.
[S56]	I. Chowdhury and M. Zulkernine, “Can Complexity, Coupling, and Cohesion Metrics Be Used as Early Indicators of Vulnerabilities?,” in Proceedings of the 2010 ACM Symposium on Applied Computing, 2010, pp. 1963–1969. doi: 10.1145/1774088.1774504.
[S57]	F. Camilo, A. Meneely, and M. Nagappan, “Do Bugs Foreshadow Vulnerabilities? A Study of the Chromium Project,” in 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories, 2015, pp. 269–279. doi: 10.1109/MSR.2015.32.
[S58]	A. Aumpansub and Z. Huang, “Detecting Software Vulnerabilities Using Neural Networks,” in 2021 13th International Conference on Machine Learning and Computing, 2021, pp. 166–171. doi: 10.1145/3457682.3457707.
[S59]	D. Zou, S. Wang, S. Xu, Z. Li, and H. Jin, “μμVulDeePecker: A Deep Learning-Based System for Multiclass Vulnerability Detection,” IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 5, pp. 2224–2236, 2021, doi: 10.1109/TDSC.2019.2942930.
[S60]	J. R. Jones, “Estimating Software Vulnerabilities,” IEEE Security & Privacy, vol. 5, no. 4, pp. 28–32, 2007, doi: 10.1109/MSP.2007.81.
[S61]	L. Yang, X. Li, and Y. Yu, “VulDigger: A Just-in-Time and Cost-Aware Tool for Digging Vulnerability-Contributing Changes,” in GLOBECOM 2017 - 2017 IEEE Global Communications Conference, 2017, pp. 1–7. doi: 10.1109/GLOCOM.2017.8254428.
[S62]	N. Ziems and S. Wu, “Security Vulnerability Detection Using Deep Learning Natural Language Processing,” in IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2021, pp. 1–6. doi: 10.1109/INFOCOMWKSHPS51825.2021.9484500.
[S63]	S. E. Şahin, E. M. Özyedierler, and A. Tosun, “Predicting vulnerability inducing function versions using node embeddings and graph neural networks,” Information and Software Technology, vol. 145, p. 106822, 2022, doi: https://doi.org/10.1016/j.infsof.2022.106822.
[S64]	I. Kalouptsoglou, M. Siavvas, D. Tsoukalas, and D. Kehagias, “Cross-project vulnerability prediction based on software metrics and deep learning,” in International Conference on Computational Science and Its Applications, 2020, pp. 877–893. doi: https://doi.org/10.1007/978-3-030-58811-3_62.
[S65]	O. H. Alhazmi and Y. K. Malaiya, “Application of Vulnerability Discovery Models to Major Operating Systems,” IEEE Transactions on Reliability, vol. 57, no. 1, pp. 14–22, 2008, doi: 10.1109/TR.2008.916872.
[S66]	J. Tian, W. Xing, and Z. Li, “BVDetector: A program slice-based binary code vulnerability intelligent detection system,” Information and Software Technology, vol. 123, p. 106289, 2020, doi: https://doi.org/10.1016/j.infsof.2020.106289.
[S67]	V. H. Nguyen and L. M. S. Tran, “Predicting Vulnerable Software Components with Dependency Graphs,” 2010. doi: 10.1145/1853919.1853923.
[S68]	D. Last, “Consensus forecasting of zero-day vulnerabilities for network security,” in 2016 IEEE International Carnahan Conference on Security Technology (ICCST), 2016, pp. 1–8. doi: 10.1109/CCST.2016.7815718.

[S69]	W. Zheng, J. Gao, X. Wu, Y. Xun, G. Liu, and X. Chen, “An Empirical Study of High-Impact Factors for Machine Learning-Based Vulnerability Detection,” in 2020 IEEE 2nd International Workshop on Intelligent Bug Fixing (IBF), 2020, pp. 26–34. doi: 10.1109/IBF50092.2020.9034888.
[S70]	S. Rahimi and M. Zargham, “Vulnerability Scrying Method for Software Vulnerability Discovery Prediction Without a Vulnerability Database,” IEEE Transactions on Reliability, vol. 62, no. 2, pp. 395–407, 2013, doi: 10.1109/TR.2013.2257052.
[S71]	A. Pechenkin and R. Demidov, “Applying Deep Learning and Vector Representation for Software Vulnerabilities Detection,” 2018. doi: 10.1145/3264437.3264489.
[S72]	N. Munaiah, F. Camilo, W. Wigham, A. Meneely, and M. Nagappan, “Do bugs foreshadow vulnerabilities? An in-depth study of the chromium project,” Empirical Software Engineering, vol. 22, no. 3, pp. 1305–1347, 2017, doi: https://doi.org/10.1007/s10664-016-9447-3.
[S73]	H. Yang, L. Ying, and L. Zhang, “Source Code Vulnerability Detection Method with Multidimensional Representation,” in International Conference on Security and Privacy in New Computing Environments, 2021, pp. 132–139. doi: https://doi.org/10.1007/978-3-030-96791-8_10.
[S74]	L. K. Shar and H. B. K. Tan, “Mining Input Sanitization Patterns for Predicting SQL Injection and Cross Site Scripting Vulnerabilities,” in Proceedings of the 34th International Conference on Software Engineering, 2012, pp. 1293–1296. doi: 10.1109/ICSE.2012.6227096.
[S75]	A. Xu, T. Dai, H. Chen, Z. Ming, and W. Li, “Vulnerability Detection for Source Code Using Contextual LSTM,” in 2018 5th International Conference on Systems and Informatics (ICSAI), 2018, pp. 1225–1230. doi: 10.1109/ICSAI.2018.8599360.
[S76]	S. Wei, H. Zhong, C. Shan, L. Ye, X. Du, and M. Guizani, “Vulnerability Prediction Based on Weighted Software Network for Secure Software Building,” in 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–6. doi: 10.1109/GLOCOM.2018.8647583.
[S77]	G. Lin et al., “Cross-Project Transfer Representation Learning for Vulnerable Function Discovery,” IEEE Transactions on Industrial Informatics, vol. 14, no. 7, pp. 3289–3297, 2018, doi: 10.1109/TII.2018.2821768.
[S78]	S. Moshtari and A. Sami, “Evaluating and Comparing Complexity, Coupling and a New Proposed Set of Coupling Metrics in Cross-Project Vulnerability Prediction,” in Proceedings of the 31st Annual ACM Symposium on Applied Computing, 2016, pp. 1415–1421. doi: 10.1145/2851613.2851777.
[S79]	P. K. Kudjo, J. Chen, S. A. Brown, and S. Mensah, “The Effect of Weighted Moving Windows on Security Vulnerability Prediction,” in 2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW), 2019, pp. 65–68. doi: 10.1109/ASEW.2019.00031.
[S80]	L. K. Shar, L. C. Briand, and H. B. K. Tan, “Web Application Vulnerability Prediction Using Hybrid Program Analysis and Machine Learning,” IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 6, pp. 688–707, 2015, doi: 10.1109/TDSC.2014.2373377.
[S81]	A. Shukla and B. Katt, “Change Point Problem in Security Vulnerability Discovery Model,” in 2019 International Conference on Software Security and Assurance (ICSSA), 2019, pp. 21–26. doi: 10.1109/ICSSA48308.2019.00010.
[S82]	K. Z. Sultana, B. J. Williams, and A. Bosu, “A Comparison of Nano-Patterns vs. Software Metrics in Vulnerability Prediction,” in 2018 25th Asia-Pacific Software Engineering Conference (APSEC), 2018, pp. 355–364. doi: 10.1109/APSEC.2018.00050.

[S83]	V. Nguyen, T. Le, O. De Vel, P. Montague, J. Grundy, and D. Phung, “Information-theoretic Source Code Vulnerability Highlighting,” in 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1–8. doi: 10.1109/IJCNN52387.2021.9533907.
[S84]	T. Zimmermann, N. Nagappan, and L. Williams, “Searching for a Needle in a Haystack: Predicting Security Vulnerabilities for Windows Vista,” in 2010 Third International Conference on Software Testing, Verification and Validation, 2010, pp. 421–428. doi: 10.1109/ICST.2010.32.
[S85]	H. Wang et al., “Combining Graph-Based Learning With Automated Data Collection for Code Vulnerability Detection,” IEEE Transactions on Information Forensics and Security, vol. 16, pp. 1943–1958, 2021, doi: 10.1109/TIFS.2020.3044773.
[S86]	S. Wei, X. Du, C. Hu, and C. Shan, “Predicting vulnerable software components using software network graph,” in International Symposium on Cyberspace Safety and Security, 2017, pp. 280–290. doi: https://doi.org/10.1007/978-3-319-69471-9_21.
[S87]	R. Scandariato, J. Walden, A. Hovsepyan, and W. Joosen, “Predicting Vulnerable Software Components via Text Mining,” IEEE Transactions on Software Engineering, vol. 40, no. 10, pp. 993–1006, 2014, doi: 10.1109/TSE.2014.2340398.
[S88]	G. Lin, W. Xiao, J. Zhang, and Y. Xiang, “Deep learning-based vulnerable function detection: A benchmark,” in International Conference on Information and Communications Security, 2019, pp. 219–232. doi: https://doi.org/10.1007/978-3-030-41579-2_13.
[S89]	A. Hovsepyan, R. Scandariato, W. Joosen, and J. Walden, “Software Vulnerability Prediction Using Text Analysis Techniques,” in Proceedings of the 4th International Workshop on Security Measurements and Metrics, 2012, pp. 7–10. doi: 10.1145/2372225.2372230.
[S90]	M. A. Albahar, “A Modified Maximal Divergence Sequential Auto-Encoder and Time Delay Neural Network Models for Vulnerable Binary Codes Detection,” IEEE Access, vol. 8, pp. 14999–15006, 2020, doi: 10.1109/ACCESS.2020.2965726.
[S91]	M. Siavvas, D. Kehagias, and D. Tzovaras, “A Preliminary Study on the Relationship Among Software Metrics and Specific Vulnerability Types,” in 2017 International Conference on Computational Science and Computational Intelligence (CSCI), 2017, pp. 916–921. doi: 10.1109/CSCI.2017.159.
[S92]	N. Munaiah and A. Meneely, “Data-Driven Insights from Vulnerability Discovery Metrics,” in 2019 IEEE/ACM Joint 4th International Workshop on Rapid Continuous Software Engineering and 1st International Workshop on Data-Driven Decisions, Experimentation and Evolution (RCoSE/DDrEE), 2019, pp. 1–7. doi: 10.1109/RCoSE/DDrEE.2019.00008.
[S93]	S. Liu et al., “CD-VulD: Cross-Domain Vulnerability Discovery Based on Deep Domain Adaptation,” IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 1, pp. 438–451, 2022, doi: 10.1109/TDSC.2020.2984505.
[S94]	K. A. Jackson and B. T. Bennett, “Locating SQL Injection Vulnerabilities in Java Byte Code Using Natural Language Techniques,” in SoutheastCon 2018, 2018, pp. 1–5. doi: 10.1109/SECON.2018.8478870.
[S95]	S. Zhang, D. Caragea, and X. Ou, “An empirical study on using the national vulnerability database to predict software vulnerabilities,” in International conference on database and expert systems applications, 2011, pp. 217–231. doi: https://doi.org/10.1007/978-3-642-23088-2_15.
[S96]	R. Russell et al., “Automated Vulnerability Detection in Source Code Using Deep Representation Learning,” in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 757–762. doi: 10.1109/ICMLA.2018.00120.

[S97]	W. Zheng, Y. Jiang, and X. Su, “Vu1SPG: Vulnerability detection based on slice property graph representation learning,” in 2021 IEEE 32nd International Symposium on Software Reliability Engineering (ISSRE), 2021, pp. 457–467. doi: 10.1109/ISSRE52982.2021.00054.
[S98]	Y. Chen, A. E. Santosa, A. M. Yi, A. Sharma, A. Sharma, and D. Lo, “A Machine Learning Approach for Vulnerability Curation,” in Proceedings of the 17th International Conference on Mining Software Repositories, 2020, pp. 32–42. doi: 10.1145/3379597.3387461.
[S99]	H. Zhang, Y. Bi, H. Guo, W. Sun, and J. Li, “ISVSF: Intelligent Vulnerability Detection Against Java via Sentence-Level Pattern Exploring,” IEEE Systems Journal, vol. 16, no. 1, pp. 1032–1043, 2022, doi: 10.1109/JSYST.2021.3072154.
[S100]	A. Challande, R. David, and G. Renault, “Building a Commit-Level Dataset of Real-World Vulnerabilities,” in Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy, 2022, pp. 101–106. doi: 10.1145/3508398.3511495.
[S101]	Y. Pang, X. Xue, and H. Wang, “Predicting Vulnerable Software Components through Deep Neural Network,” in Proceedings of the 2017 International Conference on Deep Learning Technologies, 2017, pp. 6–10. doi: 10.1145/3094243.3094245.
[S102]	D. Cao, J. Huang, X. Zhang, and X. Liu, “FTCLNet: Convolutional LSTM with Fourier Transform for Vulnerability Detection,” in 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2020, pp. 539–546. doi: 10.1109/TrustCom50675.2020.00078.
[S103]	G. Partenza, T. Amburgey, L. Deng, J. Dehlinger, and S. Chakraborty, “Automatic Identification of Vulnerable Code: Investigations with an AST-Based Neural Network,” in 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), 2021, pp. 1475–1482. doi: 10.1109/COMPSAC51774.2021.00219.
[S104]	K. Z. Sultana, B. J. Williams, and T. Bhowmik, “A study examining relationships between micro patterns and security vulnerabilities,” Software Quality Journal, vol. 27, no. 1, pp. 5–41, 2019, doi: https://doi.org/10.1007/s11219-017-9397-z.
[S105]	Z. Li, D. Zou, S. Xu, H. Jin, Y. Zhu, and Z. Chen, “SySeVR: A Framework for Using Deep Learning to Detect Software Vulnerabilities,” IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 4, pp. 2244–2258, 2022, doi: 10.1109/TDSC.2021.3051525.
[S106]	H. Alves, B. Fonseca, and N. Antunes, “Software Metrics and Security Vulnerabilities: Dataset and Exploratory Study,” in 2016 12th European Dependable Computing Conference (EDCC), 2016, pp. 37–44. doi: 10.1109/EDCC.2016.34.
[S107]	S. A. Mokhov, J. Paquet, and M. Debbabi, “The Use of NLP Techniques in Static Code Analysis to Detect Weaknesses and Vulnerabilities,” in Advances in Artificial Intelligence, 2014, pp. 326–332. doi: https://doi.org/10.1007/978-3-319-06483-3_33.
[S108]	J. Kim, Y. K. Malaiya, and I. Ray, “Vulnerability Discovery in Multi-Version Software Systems,” in 10th IEEE High Assurance Systems Engineering Symposium (HASE’07), 2007, pp. 141–148. doi: 10.1109/HASE.2007.55.
[S109]	R. Sharma, R. Sibal, and S. Sabharwal, “Change Point Modelling in the Vulnerability Discovery Process,” in Advanced Informatics for Computing Research, 2019, pp. 559–568. doi: https://doi.org/10.1007/978-981-13-3143-5_46.
[S110]	É. Leverett, M. Rhode, and A. Wedgbury, “Vulnerability Forecasting: Theory and Practice.,” Digital Threats, Mar. 2022, doi: 10.1145/3492328.
[S111]	Y. Zhang, D. Lo, X. Xia, B. Xu, J. Sun, and S. Li, “Combining Software Metrics and Text Features for Vulnerable File Prediction,” in 2015 20th International Conference on Engineering of Complex Computer Systems (ICECCS), 2015, pp. 40–49. doi: 10.1109/ICECCS.2015.15.

[S112]	A. Gkortzis, D. Feitosa, and D. Spinellis, “A Double-Edged Sword? Software Reuse and Potential Security Vulnerabilities,” in Reuse in the Big Data Era, 2019, pp. 187–203. doi: https://doi.org/10.1007/978-3-030-22888-0_13.
[S113]	Z. Yu, C. Theisen, L. Williams, and T. Menzies, “Improving Vulnerability Inspection Efficiency Using Active Learning,” IEEE Transactions on Software Engineering, vol. 47, no. 11, pp. 2401–2420, 2021, doi: 10.1109/TSE.2019.2949275.
[S114]	X. Wang, K. Sun, A. Batcheller, and S. Jajodia, “Detecting ‘0-Day’ Vulnerability: An Empirical Study of Secret Security Patch in OSS,” in 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2019, pp. 485–492. doi: 10.1109/DSN.2019.00056.
[S115]	Y. Tang, F. Zhao, Y. Yang, H. Lu, Y. Zhou, and B. Xu, “Predicting Vulnerable Components via Text Mining or Software Metrics? An Effort-Aware Perspective,” in 2015 IEEE International Conference on Software Quality, Reliability and Security, 2015, pp. 27–36. doi: 10.1109/QRS.2015.15.
[S116]	P. K. Kapur, V. S. S. Yadavali, and A. K. Shrivastava, “A comparative study of vulnerability discovery modeling and software reliability growth modeling,” in 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015, pp. 246–251. doi: 10.1109/ABLAZE.2015.7155000.
[S117]	S. E. Ponta, H. Plate, A. Sabetta, M. Bezzi, and C. Dangremont, “A Manually-Curated Dataset of Fixes to Vulnerabilities of Open-Source Software,” in Proceedings of the 16th International Conference on Mining Software Repositories, 2019, pp. 383–387. doi: 10.1109/MSR.2019.00064.
[S118]	X. Wang, R. Ma, B. Li, D. Tian, and X. Wang, “E-WBM: An Effort-Based Vulnerability Discovery Model,” IEEE Access, vol. 7, pp. 44276–44292, 2019, doi: 10.1109/ACCESS.2019.2907977.
[S119]	A. Gkortzis, D. Feitosa, and D. Spinellis, “Software reuse cuts both ways: An empirical analysis of its relationship with security vulnerabilities,” Journal of Systems and Software, vol. 172, p. 110653, 2021, doi: https://doi.org/10.1016/j.jss.2020.110653.
[S120]	G. Nikitopoulos, K. Dritsa, P. Louridas, and D. Mitropoulos, “CrossVul: A Cross-Language Vulnerability Dataset with Commit Data,” in Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2021, pp. 1565–1569. doi: 10.1145/3468264.3473122.
[S121]	T.-Y. Chong, V. Anu, and K. Z. Sultana, “Using Software Metrics for Predicting Vulnerable Code-Components: A Study on Java and Python Open Source Projects,” in 2019 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), 2019, pp. 98–103. doi: 10.1109/CSE/EUC.2019.00028.
[S122]	Z. Guan, X. Wang, W. Xin, and J. Wang, “Code Property Graph-Based Vulnerability Dataset Generation for Source Code Detection,” in Frontiers in Cyber Security, 2020, pp. 584–591. doi: https://doi.org/10.1007/978-981-15-9739-8_43.
[S123]	A. Shukla, B. Katt, and L. O. Nweke, “Vulnerability Discovery Modelling With Vulnerability Severity,” in 2019 IEEE Conference on Information and Communication Technology, 2019, pp. 1–6. doi: 10.1109/CICT48419.2019.9066187.
[S124]	W. Zheng et al., “The impact factors on the performance of machine learning-based vulnerability detection: A comparative study,” Journal of Systems and Software, vol. 168, p. 110659, 2020, doi: https://doi.org/10.1016/j.jss.2020.110659.
[S125]	Y. Wu, J. Lu, Y. Zhang, and S. Jin, “Vulnerability Detection in C/C++ Source Code With Graph Representation Learning,” in 2021 IEEE 11th Annual Computing and Communication

	Workshop and Conference (CCWC), 2021, pp. 1519–1524. doi: 10.1109/CCWC51732.2021.9376145.
[S126]	C. B. Şahin, Ö. B. Dinler, and L. Abualigah, “Prediction of software vulnerability based deep symbiotic genetic algorithms: Phenotyping of dominant-features,” Applied Intelligence, vol. 51, no. 11, pp. 8271–8287, 2021, doi: https://doi.org/10.1007/s10489-021-02324-3.
[S127]	H. Joh and Y. K. Malaiya, “Seasonal Variation in the Vulnerability Discovery Process,” in 2009 International Conference on Software Testing Verification and Validation, 2009, pp. 191–200. doi: 10.1109/ICST.2009.9.
[S128]	W. Ouyang, M. Li, Q. Liu, and J. Wang, “Binary Vulnerability Mining Based on Long Short-Term Memory Network,” in 2021 World Automation Congress (WAC), 2021, pp. 71–76. doi: 10.23919/WAC50355.2021.9559467.
[S129]	K. Kuk, P. Milić, and S. Denić, “Object-oriented software metrics in software code vulnerability analysis,” in 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), 2020, pp. 1–6. doi: 10.1109/INISTA49547.2020.9194645.
[S130]	J. Tian, J. Zhang, and F. Liu, “BBregLocator: A Vulnerability Detection System Based on Bounding Box Regression,” in 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2021, pp. 93–100. doi: 10.1109/DSN-W52860.2021.00026.
[S131]	E. Venson, T. F. Lam, B. Clark, and B. Boehm, “Analyzing Software Security-related Size and its Relationship with Vulnerabilities in OSS,” in 2021 IEEE 21st International Conference on Software Quality, Reliability and Security (QRS), 2021, pp. 956–965. doi: 10.1109/QRS54544.2021.00105.
[S132]	A. A. Elkhail and T. Cerny, “On Relating Code Smells to Security Vulnerabilities,” in 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), 2019, pp. 7–12. doi: 10.1109/BigDataSecurity-HPSC-IDS.2019.00013.
[S133]	L. Liu, L. Ci, and W. Liu, “Predicting SDC Vulnerability of Instructions Based on Random Forests Algorithm,” in International Conference on Algorithms and Architectures for Parallel Processing, 2018, pp. 593–607. doi: https://doi.org/10.1007/978-3-030-05057-3_44.
[S134]	Y. Fan, S. Shang, and X. Ding, “Smart Contract Vulnerability Detection Based on Dual Attention Graph Convolutional Network,” in Collaborative Computing: Networking, Applications and Worksharing, 2021, pp. 335–351. doi: https://doi.org/10.1007/978-3-030-92638-0_20.
[S135]	D. Zhao et al., “CVSkSA: cross-architecture vulnerability search in firmware based on kNN-SVM and attributed control flow graph,” Software Quality Journal, vol. 27, no. 3, pp. 1045–1068, 2019, doi: https://doi.org/10.1007/s11219-018-9435-5.
[S136]	N. Saccente, J. Dehlinger, L. Deng, S. Chakraborty, and Y. Xiong, “Project Achilles: A Prototype Tool for Static Method-Level Vulnerability Detection of Java Source Code Using a Recurrent Neural Network,” in 2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW), 2019, pp. 114–121. doi: 10.1109/ASEW.2019.00040.
[S137]	D. Last, “Forecasting Zero-Day Vulnerabilities,” 2016. doi: 10.1145/2897795.2897813.
[S138]	N. Medeiros, N. Ivaki, P. Costa, and M. Vieira, “Software Metrics as Indicators of Security Vulnerabilities,” in 2017 IEEE 28th International Symposium on Software Reliability Engineering (ISSRE), 2017, pp. 216–227. doi: 10.1109/ISSRE.2017.11.
[S139]	Y. Shin, A. Meneely, L. A. Williams, and J. A. Osborne, “Evaluating Complexity, Code Churn, and Developer Activity Metrics as Indicators of Software Vulnerabilities,” IEEE Transactions on Software Engineering, vol. 37, pp. 772–787, 2011, doi: 10.1109/TSE.2010.81.

[S140]	W. An, L. Chen, J. Wang, G. Du, G. Shi, and D. Meng, “AVDHRAM: Automated Vulnerability Detection based on Hierarchical Representation and Attention Mechanism,” in 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2020, pp. 337–344. doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00068.
[S141]	Y. Pang, X. Xue, and A. S. Namin, “Predicting Vulnerable Software Components through N-Gram Analysis and Statistical Feature Selection,” in 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 2015, pp. 543–548. doi: 10.1109/ICMLA.2015.99.
[S142]	Y. Pang, X. Xue, and A. S. Namin, “Early Identification of Vulnerable Software Components via Ensemble Learning,” in 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), 2016, pp. 476–481. doi: 10.1109/ICMLA.2016.0084.
[S143]	K. Z. Sultana, A. Deo, and B. J. Williams, “A Preliminary Study Examining Relationships Between Nano-Patterns and Software Security Vulnerabilities,” in 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), 2016, vol. 1, pp. 257–262. doi: 10.1109/COMPSAC.2016.34.
[S144]	Y. Mao, Y. Li, J. Sun, and Y. Chen, “Explainable Software vulnerability detection based on Attention-based Bidirectional Recurrent Neural Networks,” in 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 4651–4656. doi: 10.1109/BigData50022.2020.9377803.
[S145]	K. Filus, M. Siavvas, J. Domańska, and E. Gelenbe, “The Random Neural Network as a Bonding Model for Software Vulnerability Prediction,” in Modelling, Analysis, and Simulation of Computer and Telecommunication Systems, 2021, pp. 102–116. doi: https://doi.org/10.1007/978-3-030-68110-4_7.
[S146]	X. Chen, Y. Zhao, Z. Cui, G. Meng, Y. Liu, and Z. Wang, “Large-Scale Empirical Studies on Effort-Aware Security Vulnerability Prediction Methods,” IEEE Transactions on Reliability, vol. 69, no. 1, pp. 70–87, 2020, doi: 10.1109/TR.2019.2924932.
[S147]	Y. Zhou, S. Liu, J. Siow, X. Du, and Y. Liu, “Devign: Effective Vulnerability Identification by Learning Comprehensive Program Semantics via Graph Neural Networks,” in Proceedings of the 33rd International Conference on Neural Information Processing Systems, Red Hook, NY, USA: Curran Associates Inc., 2019. doi: https://doi.org/10.48550/arXiv.1909.03496.
[S148]	R. Lagerström, C. Baldwin, A. MacCormack, D. Sturtevant, and L. Doolan, “Exploring the Relationship Between Architecture Coupling and Software Vulnerabilities,” in Engineering Secure Software and Systems, 2017, pp. 53–69. doi: https://doi.org/10.1007/978-3-319-62105-0_4.
[S149]	F. Massacci and V. H. Nguyen, “An Empirical Methodology to Evaluate Vulnerability Discovery Models,” IEEE Transactions on Software Engineering, vol. 40, no. 12, pp. 1147–1162, 2014, doi: 10.1109/TSE.2014.2354037.
[S150]	S. Liu, M. Dibaei, Y. Tai, C. Chen, J. Zhang, and Y. Xiang, “Cyber Vulnerability Intelligence for Internet of Things Binary,” IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 2154–2163, 2020, doi: 10.1109/TII.2019.2942800.
[S151]	V. H. Nguyen and F. Massacci, “An Idea of an Independent Validation of Vulnerability Discovery Models,” in Proceedings of the 4th International Conference on Engineering Secure Software and Systems, 2012, pp. 89–96. doi: 10.1007/978-3-642-28166-2_9.

[S152]	Y. Shin and L. Williams, “Can traditional fault prediction models be used for vulnerability prediction?,” Empirical Software Engineering, vol. 18, no. 1, pp. 25–59, 2013, doi: https://doi.org/10.1007/s10664-011-9190-8.
[S153]	O. H. Alhazmi and Y. K. Malaiya, “Modeling the vulnerability discovery process,” in 16th IEEE International Symposium on Software Reliability Engineering (ISSRE’05), 2005, p. 10 pp. – 138. doi: 10.1109/ISSRE.2005.30.
[S154]	B. Chernis and R. Verma, “Machine Learning Methods for Software Vulnerability Detection,” in Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics, 2018, pp. 31–39. doi: 10.1145/3180445.3180453.
[S155]	P. Morrison, K. Herzig, B. Murphy, and L. Williams, “Challenges with Applying Vulnerability Prediction Models,” 2015. doi: 10.1145/2746194.2746198.
[S156]	J. Walden, J. Stuckman, and R. Scandariato, “Predicting Vulnerable Components: Software Metrics vs Text Mining,” in 2014 IEEE 25th International Symposium on Software Reliability Engineering, 2014, pp. 23–33. doi: 10.1109/ISSRE.2014.32.
[S157]	N. Medeiros, N. Ivaki, P. Costa, and M. Vieira, “Vulnerable Code Detection Using Software Metrics and Machine Learning,” IEEE Access, vol. 8, pp. 219174–219198, 2020, doi: 10.1109/ACCESS.2020.3041181.
[S158]	S. Chakraborty, R. Krishna, Y. Ding, and B. Ray, “Deep Learning Based Vulnerability Detection: Are We There Yet?,” IEEE Transactions on Software Engineering, vol. 48, no. 9, pp. 3280–3296, 2022, doi: 10.1109/TSE.2021.3087402.
[S159]	J. Yang, D. Ryu, and J. Baik, “Improving vulnerability prediction accuracy with Secure Coding Standard violation measures,” in 2016 International Conference on Big Data and Smart Computing (BigComp), 2016, pp. 115–122. doi: 10.1109/BIGCOMP.2016.7425809.
[S160]	C. Catal, A. Akbulut, E. Ekenoglu, and M. Alemdaroglu, “Development of a Software Vulnerability Prediction Web Service Based on Artificial Neural Networks ,” in Trends and Applications in Knowledge Discovery and Data Mining, 2017, pp. 59–67. doi: https://doi.org/10.1007/978-3-319-67274-8_6.
[S161]	J. Stuckman, J. Walden, and R. Scandariato, “The Effect of Dimensionality Reduction on Software Vulnerability Prediction Models,” IEEE Transactions on Reliability, vol. 66, no. 1, pp. 17–37, 2017, doi: 10.1109/TR.2016.2630503.
[S162]	H. Sun, Y. Tong, J. Zhao, and Z. Gu, “DVul-WLG: Graph Embedding Network Based on Code Similarity for Cross-Architecture Firmware Vulnerability Detection,” in Information Security, 2021, pp. 320–337. doi: https://doi.org/10.1007/978-3-030-91356-4_17.
[S163]	P. K. Kudjo, S. B. Aformaley, S. Mensah, and J. Chen, “The Significant Effect of Parameter Tuning on Software Vulnerability Prediction Models,” in 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2019, pp. 526–527. doi: 10.1109/QRS-C.2019.00107.
[S164]	B. M. Padmanabhuni and H. B. K. Tan, “Buffer Overflow Vulnerability Prediction from x86 Executables Using Static Analysis and Machine Learning,” in 2015 IEEE 39th Annual Computer Software and Applications Conference, 2015, vol. 2, pp. 450–459. doi: 10.1109/COMPSAC.2015.78.
[S165]	I. Abunadi and M. Alenezi, “Towards Cross Project Vulnerability Prediction in Open Source Web Applications,” 2015. doi: 10.1145/2832987.2833051.
[S166]	P. Sun, L. Garcia, G. Salles-Loustau, and S. Zonouz, “Hybrid Firmware Analysis for Known Mobile and IoT Security Vulnerabilities,” in 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2020, pp. 373–384. doi: 10.1109/DSN48063.2020.00053.

[S167]	O. H. Alhazmi and Y. K. Malaiya, “Prediction capabilities of vulnerability discovery models,” in RAMS ’06. Annual Reliability and Maintainability Symposium, 2006., 2006, pp. 86–91. doi: 10.1109/RAMS.2006.1677355.
[S168]	B. Mosolygó, N. Vándor, G. Antal, P. Hegedűs, and R. Ferenc, “Towards a Prototype Based Explainable JavaScript Vulnerability Prediction Model,” in 2021 International Conference on Code Quality (ICCQ), 2021, pp. 15–25. doi: 10.1109/ICCQ51190.2021.9392984.
[S169]	F. K. Wai, L. W. Yong, D. M. Divakaran, and V. L. L. Thing, “Predicting vulnerability discovery rate using past versions of a software,” in 2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), 2018, pp. 220–225. doi: 10.1109/SOLI.2018.8476753.
[S170]	V. Nguyen et al., “Deep Domain Adaptation for Vulnerable Code Function Identification,” in 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1–8. doi: 10.1109/IJCNN.2019.8851923.
[S171]	I. Chowdhury and M. Zulkernine, “Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities,” Journal of Systems Architecture, vol. 57, no. 3, pp. 294–313, 2011, doi: https://doi.org/10.1016/j.sysarc.2010.06.003.
[S172]	K. Z. Sultana, Z. Codabux, and B. Williams, “Examining the Relationship of Code and Architectural Smells with Software Vulnerabilities,” in 2020 27th Asia-Pacific Software Engineering Conference (APSEC), 2020, pp. 31–40. doi: 10.1109/APSEC51365.2020.00011.
[S173]	W. Han, J. Pang, X. Zhou, and D. Zhu, “Binary software vulnerability detection method based on attention mechanism,” in 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), 2020, pp. 1462–1466. doi: 10.1109/ICMCCE51767.2020.00320.
[S174]	J. Zhou et al., “Finding A Needle in a Haystack: Automated Mining of Silent Vulnerability Fixes,” in 2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2021, pp. 705–716. doi: 10.1109/ASE51524.2021.9678720.
[S175]	C. B. Şahin, “DCW-RNN: Improving Class Level Metrics for Software Vulnerability Detection Using Artificial Immune System with Clock-Work Recurrent Neural Network,” in 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), 2021, pp. 1–8. doi: 10.1109/INISTA52262.2021.9548609.
[S176]	K. Z. Sultana and B. J. Williams, “Evaluating micro patterns and software metrics in vulnerability prediction,” in 2017 6th International Workshop on Software Mining (SoftwareMining), 2017, pp. 40–47. doi: 10.1109/SOFTWAREMINING.2017.8100852.
[S177]	P. K. Kudjo and J. Chen, “A Cost-Effective Strategy for Software Vulnerability Prediction Based on Bellwether Analysis,” in Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis, 2019, pp. 424–427. doi: 10.1145/3293882.3338985.
[S178]	Z. Bilgin, M. A. Ersoy, E. U. Soykan, E. Tomur, P. Çomak, and L. Karaçay, “Vulnerability Prediction From Source Code Using Machine Learning,” IEEE Access, vol. 8, pp. 150672–150684, 2020, doi: 10.1109/ACCESS.2020.3016774.
[S179]	H. K. Dam, T. Tran, T. Pham, S. W. Ng, J. Grundy, and A. Ghose, “Automatic Feature Learning for Predicting Vulnerable Software Components,” IEEE Transactions on Software Engineering, vol. 47, no. 1, pp. 67–85, 2021, doi: 10.1109/TSE.2018.2881961.
[S180]	A. Fidalgo, I. Medeiros, P. Antunes, and N. Neves, “Towards a Deep Learning Model for Vulnerability Detection on Web Application Variants,” in 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 2020, pp. 465–476. doi: 10.1109/ICSTW50294.2020.00083.