Universidade Federal de Goiás

Instituto de Informática

Algoritmos de aprendizado de máquina e vulnerabilidades em softwares: uma revisão sistemática

Isabela Teixeira e Thiago de Souza

Introdução

Inicialmente deve-se definir o que são vulnerabilidades em softwares: são defeitos no design ou implementação no código de softwares que podem ser exploradas por indivíduos mal-intencionados levando ao comprometimento da segurança e integridade do software, podendo ocasionar execução de ações não autorizadas, acesso não autorizado a dados, roubo de informações confidenciais, interrupção de serviços ou outras atividades prejudiciais.

Os algoritmos de inteligência artificial são algoritmos que trabalham com aprendizado de máquina, sejam supervisionados, com dados rotulados, ou não supervisionados, com dados não rotulados. Sendo possível aprender padrões e realizar previsões a partir de uma base de dados.

O objetivo desta revisão é responder se algoritmos de inteligência artificial conseguem identificar vulnerabilidades em softwares por meio de análise de código (white-box) ou pelo acesso ao software assim como faria o usuário comum (black-box).

Dessa forma, com a identificação de vulnerabilidades em softwares sendo mais eficiente seria possível reduzir e/ou mitigar as consequências de crimes cibernéticos oriundos de exploração de vulnerabilidades em softwares.

Questão de Pesquisa

Algoritmos de aprendizado de máquina podem ser usados para identificar vulnerabilidades em softwares?

Bases de Busca

- www.periodicos.capes.gov.br
- www.scopus.com
- www.webofscience.com
- www.sciencedirect.com
- www.ieexplore.com

Palavras-Chave

“Artificial Intelligence”

“Vulnerability detection”

“Software”

“Algorithms”

“code analysis”

Strings de Busca

ScienceDirect - “Artificial Intelligence” AND “Vulnerability detection” AND “Software” AND “Algorithms” AND "code analysis"

Scopus - ALL("Artificial Intelligence" AND "Vulnerability detection" AND "Software" AND "Algorithms") AND TITLE(("vulnerability" OR "vulnerabilities") AND ("detection" OR "detect" OR "spotting")) AND ABS("neural network" OR "machine learning" OR "artificial intelligence") AND PUBYEAR > 2017 AND PUBYEAR < 2024 AND (LIMIT-TO (OA,"all"))

AND (LIMIT-TO (DOCTYPE,"ar")) AND (LIMIT-TO (LANGUAGE,"English") OR LIMIT-TO (LANGUAGE,"Portuguese"))

Web of Science - (TI=(((vulnerability OR vulnerabilities) AND (detection OR detect OR spotting)))) AND AB=(("neural network" OR "machine learning" OR "artificial intelligence" OR "deep learning" OR "transfer learning") AND (algorithm OR algorithms OR method) AND (vulnerability OR vulnerabilities) AND code)

IEE xplore - (("Document Title": vulnerabilit* AND detect*) AND ("Abstract":neural network OR machine learning OR artificial intelligence))

Capes - “Artificial Intelligence” AND “Vulnerability detection” AND “Software” AND “Algorithms”

Idiomas dos Artigos

Inglês. Não foram achados artigos referentes ao tema específico em português.

Critérios de Inclusão e Exclusão Dos Trabalhos

Critérios de inclusão:

1. Período de publicação 2018 - 2023.
2. Revisados por pares.
3. Publicação do tipo artigo.
4. Artigos de acesso livre.

Critérios de exclusão:

1. Qualquer idioma diferente de Português e Inglês.
2. Artigos repetidos achados nas diferentes bases de dados.

Processo de Seleção dos Estudos

Leitura do título e abstract analisando a relevância do artigo para a questão de pesquisa.

Selecionados:

An, J.H., Wang, Z. & Joe, I. **A CNN-based automatic vulnerability detection.** J Wireless Com Network 2023, 41 (2023). https://doi.org/10.1186/s13638-023-02255-2

Cho Do Xuan (2023) **A new approach to software vulnerability detection based on CPG analysis**, Cogent Engineering, 10:1, DOI: 10.1080/23311916.2023.2221962

Al-Boghdady, Abdullah & El-Ramly, Mohammad & Wassif, Khaled. (2022). **iDetect for vulnerability detection in internet of things operating systems using machine learning**. Scientific Reports. 12. 10.1038/s41598-022-21325-x.

Zhou, Xin & Pang, Jianmin & Yue, Feng & Liu, Fudong & Guo, Jiayu & Liu, Wenfu& Song, Zhihui & Shu, Guoqiang & Xia, Bing & Shan, Zheng. (2022). **A new method of software vulnerability detection based on a quantum neural network**. Scientific Reports. 12. 10.1038/s41598-022- 11227-3.

Li, Zhen & Zou, Deqing & Xu, Shouhuai & Jin, Hai & Zhu, Yawei & Chen, Zhaoxuan. (2021). **SySeVR: A Framework for Using Deep Learning to Detect Software Vulnerabilities**. IEEE Transactions on Dependable and Secure Computing. PP. 1-1. 10.1109/TDSC.2021.3051525.

Hassan, Md Maruf & Ahmad, R.Badlishah & Ghosh, Tonmoy. (2021). **SQL Injection Vulnerability Detection Using Deep Learning**: A Feature-based Approach. Indonesian Journal of Electrical Engineering and Informatics(IJEEI). 9. 10.52549/.v9i3.3131.

Wang, Huanting & Ye, Guixin & Tang, Zhanyong & Tan, Shin Hwei & Huang, Songfang & Fang, Dingyi & Feng, Yansong & Bian, Lizhong & Wang, Zheng. (2020). **Combining Graph-Based Learning With Automated Data Collection for Code Vulnerability Detection**. IEEE Transactions on Information Forensics and Security. PP. 10.1109/TIFS.2020.3044773.

Zagane, Mohammed & Abdi, Mustapha & Alenezi, Mamdouh. (2020). **Deep Learning for Software Vulnerabilities Detection Using Code Metrics**. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.2988557.

Lin, C.; Xu, Y.; Fang, Y.; Liu, Z. **VulEye: A Novel Graph Neural Network Vulnerability Detection Approach for PHP Application**. Appl. Sci. 2023, 13, 825. https://doi.org/10.3390/app13020825

Wang, L.; Li, X.; Wang, R.; Xin, Y.; Gao, M.; Chen, Y. **PreNNsem: A Heterogeneous Ensemble Learning Framework for Vulnerability Detection in Software**. Appl. Sci. 2020, 10, 7954. https://doi.org/10.3390/app10227954

Wartschinski, Laura & Noller, Yannic & Vogel, Thomas & Kehrer, Timo & Grunske, Lars. (2022). **VUDENC: Vulnerability Detection with Deep Learning on a Natural Codebase for Pytho**n.

Li, Xin & Wang, Lu & Xin, Yang & Yang, Yixian & Tang, Qifeng & Chen, Yuling. (2021). **Automated Software Vulnerability Detection Based on Hybrid Neural Network**. Applied Sciences. 11. 3201. 10.3390/app11073201.

Guo, J., Wang, Z., Li, H. et al. **Detecting vulnerability in source code using CNN and LSTM network**. Soft Comput 27, 1131–1141 (2023). https://doi.org/10.1007/s00500-021-05994-w

Qiang, G. (2022). **Research on Software VulnerabilityDetection Method Based on Improved CNN Model. Scientific Programming**.

Song, Zihua & Wang, Junfeng & Liu, Shengli & Zhiyang, Fang & Yang, Kaiyuan. (2022). **HGVul: A Code Vulnerability Detection Method Based on Heterogeneous Source-Level Intermediate Representation**. Security and Communication Networks. 2022. 1-13. 10.1155/2022/1919907.

Marjanov, Tina & Pashchenko, Ivan & Massacci, Fabio. (2022). **Machine Learning for Source Code Vulnerability Detection: What Works and What Isn't There Yet**. IEEE Security & Privacy. PP. 2-18. 10.1109/MSEC.2022.3176058.

Zolanvari, Maede & Teixeira, Marcio & Gupta, Lav & Khan, Khaled & Jain, Raj. (2019). **Machine Learning Based Network Vulnerability Analysis of Industrial Internet of Thing**s. IEEE Internet of Things Journal. PP. 1-1. 10.1109/JIOT.2019.2912022.

Xiaojun Ren, Yongtang Wu, Jiaqing Li, Dongmin Hao, Muhammad Alam. **Smart contract vulnerability detection based on a semantic code structure and a self-designed neural network**. Computers and Electrical Engineering (2023). https://doi.org/10.1016/j.compeleceng.2023.108766.

M., Hariharan & C., Sathish & Tanwar, Anshul & Sundaresan, Krishna & Ganesan, Prasanna & Ravi, Sriram & Ramamurthy, Karthik. (2022). **Proximal Instance Aggregator networks for explainable security vulnerability detection**. Future Generation Computer Systems. 134. 10.1016/j.future.2022.04.008.

Zhonglin Liu, Yong Fang, Cheng Huang, and YijiaXu. 2023. **MFXSS: An effective XSS vulnerability detection method in JavaScript based on multi-feature model**. Comput. Secur. 124, C (Jan 2023). https://doi.org/10.1016/j.cose.2022.103015

Katarzyna Filus and Joanna Domańska. 2023. **Software vulnerabilities in TensorFlow-based deep learning applications**. Comput. Secur. 124, C (Jan 2023). https://doi.org/10.1016/j.cose.2022.102948

Tian, Junfeng & Xing, Wenjing& Li, Zhen. (2020). **BVDetector: A Program Slice-based Binary Code Vulnerability IntelligentDetection System**. Information and Software Technology. 123. 106289. 10.1016/j.infsof.2020.106289.

S. Calzavara, M. Conti, R. Focardi, A. Rabitti and G. Tolomei, **"Mitch: A Machine Learning Approach to the Black-Box Detection of CSRF Vulnerabilities"**, 2019 IEEE European Symposium on Security and Privacy (EuroS&P), Stockholm, Sweden, 2019.

Wei, Hongwei & Lin, Guanjun & Li, Lin & Jia, Heming. (2021). **A Context-Aware Neural Embedding for Function-Level Vulnerability Detection**. Algorithms. 14. 10.3390/a14110335.

Abdalla Wasef Marashdih, Zarul Fitri Zaaba, Khaled Suwais. **Predicting input validation vulnerabilities based on minimal SSA features and machine learning**. Journal of King Saud University - Computer and Information Sciences (2022). https://doi.org/10.1016/j.jksuci.2022.09.010.

Kudo, Takanori & Kimura, Tomotaka & Inoue, Yoshiaki & Aman, Hirohisa & Hirata, Kouji. (2018). **Stochastic Modeling of Self-Evolving Botnets with Vulnerability Discovery**. Computer Communications. 124. 10.1016/j.comcom.2018.04.010.

Soyoung Lee, Seongil Wi, and Sooel Son. 2022. **Link: Black-Box Detection of Cross-Site Scripting VulnerabilitiesUsing Reinforcement Learning**. In Proceedings of the ACM Web Conference 2022 (WWW '22). Association for Computing Machinery, New York, NY, USA, 743–754. https://doi.org/10.1145/3485447.3512234

ŞAHİN, CANAN. (2022). **Semantic-based Vulnerability Detection by Functional Connectivity of Gated Graph Sequence Neural Networks**. 10.21203/rs.3.rs-1364332/v1.

Batur Şahin, Canan & Dinler, Özlem & Abualigah, Laith. (2021). **Prediction of software vulnerability-based deep symbiotic genetic algorithms**: Phenotyping of dominant-features. Applied Intelligence. 10.1007/s10489-021-02324-3.

Artigos não selecionados:

Lomio, Francesco & Iannone, Emanuele & Lucia, Andrea & Palomba, Fabio & Lenarduzzi, Valentina. (2022). **Just-in-Time Software Vulnerability Detection: Are We There Yet?. Journal of Systems and Software**. 188. 10.1016/j.jss.2022.111283.

Jinxiong Zhao, Sensen Guo, and Dejun Mu. 2021. **DouBiGRU-A: Software defect detection algorithm based on attention mechanism and double BiGRU**. Comput. Secur. 111, C (Dec 2021). https://doi.org/10.1016/j.cose.2021.102459

Dawei Yuan, Xiaohui Wang, Yao Li, and Tao Zhang. 2023. **Optimizing smart contract vulnerability detection via multi-modality code and entropy embedding**. J. Syst. Softw. 202, C (Aug 2023). https://doi.org/10.1016/j.jss.2023.111699

Yan Wang, Peng Jia, Xi Peng, Cheng Huang, and Jiayong Liu. 2023. **BinVulDet: Detecting vulnerability in binary program via decompiled pseudo code and BiLSTM-attention**. Comput. Secur. 125, C (Feb 2023). https://doi.org/10.1016/j.cose.2022.103023

Altarawy, Doaa & Shahin, Hossameldin & Mohammed, Ayat & Meng, Na. (2018). **Lascad : Language-Agnostic Software Categorization and Similar Application Detection**. Journal of Systems and Software. 142. 10.1016/j.jss.2018.04.018.

Navdeep S. Chahal, Preeti Bali, Praveen Kumar Khosla. **A Proactive Approach to assess web application security through the integration of security tools in a Security Orchestration Platform**. Computers & Security. 2022. https://doi.org/10.1016/j.cose.2022.102886.

Xiaozhou Li, Sergio Moreschini, Zheying Zhang, Fabio Palomba, and Davide Taibi. 2023. **The anatomy of a vulnerability database: A systematic mapping study**. J. Syst. Softw. 201, C (Jul 2023). https://doi.org/10.1016/j.jss.2023.111679

Ji, Tiantian & Zhongru, Wang & Tian, Zhihong & Fang, Binxing & Ruan, Qiang & Wang, Haichen & Shi, Wei. (2020). **AFLPro: Direction sensitive fuzzing. Journal of Information Security and Applications**. 54. 102497. 10.1016/j.jisa.2020.102497.

Mark A. Williams, Roberto Camacho Barranco, Sheikh Motahar Naim, Sumi Dey, M. Shahriar Hossain, Monika Akbar. **A vulnerability analysis and prediction framework**. Computers & Security. 2020. https://doi.org/10.1016/j.cose.2020.101751.

Attia Qamar, Ahmad Karim, and Victor Chang. 2019. **Mobile malware attacks: Review, taxonomy & future directions**. Future Gener. Comput. Syst. 97, C (Aug 2019), 887–909. https://doi.org/10.1016/j.future.2019.03.007

Sefa Eren Şahin, Ecem Mine Özyedierler, and Ayse Tosun. 2022. **Predicting vulnerability inducing function versions using node embeddings and graph neural networks**. Inf. Softw. Technol. 145, C (May 2022). https://doi.org/10.1016/j.infsof.2022.106822

Yasasin, Emrah & Prester, Julian & Wagner, Gerit & Schryen, Guido. (2019). **Forecasting IT Security Vulnerabilities - An Empirical Analysis**. Computers & Security. 88. 10.1016/j.cose.2019.101610.

Batur Şahin, C., Abualigah, L. **A novel deep learning-based feature selection model for improving the static analysis of vulnerability detection**. Neural Comput & Applic 33, 14049–14067 (2021). https://doi.org/10.1007/s00521-021-06047-x

Lejun Zhang, Jinlong Wang, Weizheng Wang, Zilong Jin, Yansen Su, and Huiling Chen. 2022. **Smart contract vulnerability detection combined with multi-objective detection**. Comput. Netw. 217, C (Nov 2022). https://doi.org/10.1016/j.comnet.2022.109289

Qianchong Zhao, Cheng Huang, Liuhu Dai. **VULDEFF: Vulnerability detection method based on function fingerprints and code differences**. Knowledge-Based Systems. 2023. https://doi.org/10.1016/j.knosys.2022.110139.

Gupta, Aakanshi & Sharma, Deepanshu & Phulli, Kritika. (2022). **ANN Modelling on Vulnerabilities Detection in Code Smells-Associated Android Applications**. Foundations of Computing and Decision Sciences. 47. 3-26. 10.2478/fcds-2022-0001.

Thilo Hagendorff. 2021. **Forbidden knowledge in machine learning reflections on the limits of research and publication**. AI Soc. 36, 3 (Sep 2021), 767–781. https://doi.org/10.1007/s00146-020-01045-4

Ahakonye, Love & Nwakanma, Cosmas & Lee, Jae Min & Kim, Dong-Seong. (2021). **Efficient Classification of Enciphered SCADA Network Traffic in Smart Factory Using Decision Tree Algorithm**. IEEE Access. 9. 154892-154901. 10.1109/ACCESS.2021.3127560.

Kusyk, Janusz & Uyar, M. & Sahin, Cem Safak. (2018). **Survey on evolutionary computation methods for cybersecurity of mobile ad hoc networks**. Evolutionary Intelligence. 10. 10.1007/s12065-018-0154-4.

Critérios de Qualidade dos Estudos

Algumas das perguntas que podem ser feitas para analisar a qualidade dos artigos são as seguintes:

Os cálculos estatísticos de eficiência dos algoritmos foram feitos de maneira criteriosa e correta?

Foi usado mais de um data-set? Caso sim, os data-sets usados são bons?

Nos estudos que propõem soluções para uma gama diversa de tecnologias foram testados diferentes variações dessa tecnologia?

Caso seja um algoritmo que o resultado dependa de poder computacional, as comparações com outros métodos foram feitas em mesmas condições? Foram usadas várias condições diferentes para os diferentes métodos propostos?

Também é importante ressaltar que o número de citações pode ser uma boa métrica para avaliar o impacto e qualidade do artigo, mas não de forma absoluta pois fatores como a popularidade do tópico, o tamanho da comunidade de pesquisadores na área e a disponibilidade do artigo para leitura podem facilmente influenciar esse número.

Estratégia de Extração de Informação

Para responder a questão de pesquisa foram lidos a introdução e a conclusão de todos os artigos selecionados. A leitura foi feita atentando-se aos seguintes tópicos:

- Qual tipo de algoritmo o artigo propõe para identificar vulnerabilidades.
- Qual ou quais tipos de vulnerabilidades o algoritmo propõe identificar.
- O algoritmo de aprendizado é supervisionado ou não supervisionado.
- Se de acordo com o artigo, o algoritmo foi eficaz para o que foi proposto.
- Se a eficiência do algoritmo foi comparada com outros métodos ou algoritmos.

Com as informações conseguidas desses tópicos de cada artigo pode-se tirar conclusões a respeito do tema e responder decentemente a questão de pesquisa.

Sumarização dos Resultados

Pode-se afirmar que os algoritmos de aprendizado de máquina têm mostrado grande potencial para identificar vulnerabilidades em softwares. Através da análise de código, padrões e comportamentos suspeitos, esses algoritmos conseguem automatizar e acelerar o processo de detecção de vulnerabilidades, contribuindo para a melhoria da segurança de sistemas.

Ao longo desta revisão, observamos uma variedade de abordagens e técnicas empregadas pelos estudos analisados, cada uma com suas vantagens e limitações. Foram constatados vários artigos que possuem propostas de algoritmos com resultados satisfatórios para vários tipos de softwares diferentes, como por exemplo, Smart contract, qualquer código em C, Python e PHP, JavaScript, programas em binário, etc. Enquanto alguns estudos se concentraram em detectar vulnerabilidades conhecidas, outros exploraram o potencial de identificar novos padrões e comportamentos maliciosos. Destaca-se, que a maioria dos algoritmos propostos utiliza redes neurais, enquanto propostas de algoritmos de outras áreas são menos citadas. Além disso, foi observado que há poucas propostas no contexto black-box, com apenas dois artigos encontrados.

Embora os resultados tenham sido satisfatórios, é importante destacar que nenhum método é totalmente infalível. A abordagem ideal deve combinar a utilização de algoritmos de aprendizado de máquina com a análise manual e a expertise de especialistas em segurança de software. Dessa forma, podemos obter uma detecção mais precisa e abrangente de

vulnerabilidades, proporcionando uma camada adicional de proteção contra ameaças cibernéticas.

Diante dos avanços e promessas mostrados pelas pesquisas nesta área, acreditamos que o uso de algoritmos de aprendizado de máquina continuará a desempenhar um papel fundamental na identificação e prevenção de vulnerabilidades em softwares, contribuindo para a construção de sistemas mais seguros e resilientes no cenário da segurança. No entanto, novas pesquisas são necessárias para aprimorar ainda mais essas técnicas e abordagens, buscando soluções mais eficazes e adaptáveis.