

- Jan (2003), 993–1022.
- [8] Steven Burrows and Seyed MM Tahaghoghi. 2007. Source code authorship attribution using n-grams. In *Proceedings of the Twelfth Australasian Document Computing Symposium, Melbourne, Australia, RMIT University*. Citeseer, 32–39.
 - [9] Steven Burrows, Alexandra L Uitdenbogerd, and Andrew Turpin. 2009. Application of information retrieval techniques for source code authorship attribution. In *International Conference on Database Systems for Advanced Applications*. Springer, 699–713.
 - [10] Steven Burrows, Alexandra L Uitdenbogerd, and Andrew Turpin. 2014. Comparing techniques for authorship attribution of source code. *Software: Practice and Experience* 44, 1 (2014), 1–32.
 - [11] Aylin Caliskan-Islam, Richard Harang, Andrew Liu, Arvind Narayanan, Clare Voss, Fabian Yamaguchi, and Rachel Greenstadt. 2015. De-anonymizing programmers via code stylometry. In *24th USENIX Security Symposium (USENIX Security)*, Washington, DC.
 - [12] Aylin Caliskan-Islam, Fabian Yamaguchi, Edwin Dauber, Richard Harang, Konrad Rieck, Rachel Greenstadt, and Arvind Narayanan. 2015. When coding style survives compilation: De-anonymizing programmers from executable binaries. *arXiv preprint arXiv:1512.08546* (2015).
 - [13] Edwin Dauber, Aylin Caliskan-Islam, Richard Harang, and Rachel Greenstadt. 2017. Git Blame Who?: Stylistic Authorship Attribution of Small, Incomplete Source Code Fragments. *arXiv preprint arXiv:1701.05681* (2017).
 - [14] Haibiao Ding and Mansur H Samadzadeh. 2004. Extraction of Java program fingerprints for software authorship identification. *Journal of Systems and Software* 72, 1 (2004), 49–57.
 - [15] Jeanne Ferrante, Karl J Ottenstein, and Joe D Warren. 1987. The program dependence graph and its use in optimization. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 9, 3 (1987), 319–349.
 - [16] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter* 11, 1 (2009), 10–18.
 - [17] RI Kilgour, AR Gray, PJ Sallis, and SG MacDonell. 1998. A fuzzy logic approach to computer software source code authorship analysis. (1998).
 - [18] Sangkyum Kim, Hyungsul Kim, Tim Weninger, Jiawei Han, and Hyun Duk Kim. 2011. Authorship classification: a discriminative syntactic tree mining approach. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 455–464.
 - [19] Jay Kothari, Maxim Shevertalov, Edward Stehle, and Spiros Mancoridis. 2007. A probabilistic approach to source code authorship identification. In *Information Technology, 2007. ITNG'07. Fourth International Conference on*. IEEE, 243–248.
 - [20] Ivan Krsul and Eugene H Spafford. 1997. Authorship analysis: Identifying the author of a program. *Computers & Security* 16, 3 (1997), 233–257.
 - [21] Thomas J McCabe. 1976. A complexity measure. *IEEE Transactions on software Engineering* 4 (1976), 308–320.
 - [22] Xiaozhu Meng and Barton P Miller. [n. d.]. Binary Code Multi-Author Identification in Multi-Toolchain Scenarios. ([n. d.]).
 - [23] Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. 2012. On the feasibility of internet-scale author identification. In *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 300–314.
 - [24] Baishakhi Ray, Daryl Posnett, Vladimir Filkov, and Premkumar Devanbu. 2014. A large scale study of programming languages and code quality in github. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM, 155–165.
 - [25] Nathan Rosenblum, Xiaojin Zhu, and Barton Miller. 2011. Who wrote this code? identifying the authors of program binaries. *Computer Security—ESORICS 2011* (2011), 172–189.
 - [26] Philip Sallis, Asbjorn Aakjaer, and Stephen MacDonell. 1996. Software forensics: old methods for a new science. In *Software Engineering: Education and Practice, 1996. Proceedings. International Conference*. IEEE, 481–485.
 - [27] Eugene H Spafford and Stephen A Weeber. 1993. Software forensics: Can we track code to its authors? *Computers & Security* 12, 6 (1993), 585–595.
 - [28] Danny van Bruggen. 2017. JavaParser. Retrieved November 15, 2017 from <https://javaparser.org/index.html>
 - [29] Chenxi Wang and John Knight. 2001. *A security architecture for survivability mechanisms*. University of Virginia.
 - [30] Mark Weiser. 1981. Program slicing. In *Proceedings of the 5th international conference on Software engineering*. IEEE Press, 439–449.
 - [31] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. 2016. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.