Research Proposal

Dr. Phuong Nguyen, Associate Professor, DISIM, University of L'Aquila, Italy
Email: phuong.nguyen@univaq.it

January 20, 2026

1 General Information

- Title: *"Detoxing Large Language Models: A Machine Unlearning Approach to Remove Malicious Code."*
- Principal Investigator: Phuong Thanh Nguyen.
- Position: Associate Professor, University of L'Aquila.
- Type of projects: Progetti di ricerca.
- ERC research sector: PE6_3 and PE6_7.
- Other members of the group: Dr. Juri Di Rocco, RTD/b, University of L'Aquila. Two master/PhD students.

2 Abstract

Large Language Models (LLMs) have been applied in Software Engineering, yielding promising performance in various tasks, including code generation and vulnerability detection. Nevertheless, their reliance on massive pretraining data raises concerns about data memorization, particularly of sensitive security datasets, and the extent to which this may bias evaluations. LLMs trained on public code repositories inadvertently memorize and regenerate malicious patterns, e.g., backdoors, exploits, or hardcoded credentials, posing severe risks when deployed in code-generation assistants. Traditional mitigation via filtering or Reinforcement Learning from Human Feedback is costly and incomplete. Thus, it is crucial to conceptualize more effective and efficient ways to deal with the issue. In this work, we propose machine unlearning as an efficient and verifiable solution to selectively erase malicious code knowledge from code-focused LLMs, while preserving benign performance.

3 State-of-the-art research

LLMs have revolutionized software automation, improving several processes including code completion, translation, and repair [13]. Despite these advancements, practitioners depend on large public code corpora for intent-aligned training, a reliance that embeds bugs, security flaws, and licensing risks into resulting LLMs [3]. As open-source software adoption accelerates, escalating supply-chain security risks have drawn significant scrutiny [21], and because LLMs might have been trained on low-quality, uncertain-source code, their outputs often carry bugs, vulnerabilities, and licensing violations that propagate into downstream models [11]. Platforms such as GitHub and Gitee provide many reusable packages for user convenience, but no security guarantees are offered for uploaded content. Thus, attackers can publish malicious libraries that developers inadvertently import, enabling backdoors and data theft [7, 17]. Pre-training data poisoning introduces an even more covert threat, arising from adversarial manipulation of openly accessible web resources. Existing work [18] showed that even marginal contamination rates translate into vast amounts of corrupted data in large-scale training pipelines.

Given the limited attention to robust mitigation strategies, advancing methodologies for malicious code removal emerges as a critically urgent and timely research issue. Essentially, it is necessary to conceptualize methods to protect LLMs from the memorization of malicious code. In other words, *detoxing LLMs* is a crucial task to ensure safe development.

Despite this urgency, existing approaches remain limited in malicious data detection and control while lacking comprehensive mechanisms to prevent contamination or restore models that have been poisoned [10]. A potential method to tackle the issue is machine unlearning [2]—a methodology designed to selectively erase specific knowledge from trained models. While extensively studied for removing sensitive information (e.g., copyrighted content) in privacy applications [1], the available benchmarks and techniques focus on natural language data [16]. Thus, this technique remains underexplored in Software Engineering (SE), particularly for the removal of malicious code patterns from poisoned models. To the best of our knowledge, there has been no approach proposed to detox LLMs dedicated to code generation tasks.

4 Methodology

Gradient Ascent (GA). To eliminate malicious code from LLMs, we employ a method to invert the original training objective. Rather than minimizing the negative log-likelihood as in conventional training, the unlearning approach maximizes the following loss function: $\mathcal{L}_{unlearn}(\theta; s) = -\sum_{j=1}^{|s|} \log P_\theta(s_j \mid s_{<j})$ where $s = (s_1, s_2, ..., s_{|s|})$ represents the token sequence to be removed, $s_{<j} = (s_1, ..., s_{j-1})$ denotes the prefix tokens, and $P_\theta(s_j \mid s_{<j})$ indicates the model's probability of predicting token s_j conditioned on the preceding tokens, with model parameters θ [12].

Gradient Ascent with Retain Set Optimization (GA+DA). Pure gradient ascent often suffers from over-forgetting as the model unlearns target sequences, it may inadvertently erase useful knowledge beyond the intended scope. To mitigate this we adopt a hybrid unlearning strategy that jointly optimizes forgetting and retention objectives. This balances the removal of malicious code patterns, while preserving model performance on benign data, as commonly adopted in prior work on approximate unlearning [20].

Let $\mathcal{D}_{forget}$ be the set of malicious samples for removal; $\mathcal{D}_{retain}$ denote a carefully selected subset of benign code from the original training corpus, $\mathcal{D}_{retain} \subseteq \mathcal{D}_{training} \setminus \mathcal{D}_{forget}$. The unlearning framework employs a dual-objective formulation to balance competing optimization goals:

$$\mathcal{L}_{unlearn}(\theta) = \sum_{w \in \mathcal{D}_{forget}} \sum_{t=1}^{T} \mathbb{E}_{q_t \sim Q_{w_t}} \log P_\theta(q_t \mid w_1, w_2, \ldots, w_{t-1}) + \sum_{z \in \mathcal{D}_{retain}} \sum_{t=1}^{T} \log P_\theta(z_t \mid z_1, z_2, \ldots, z_{t-1})$$

in which θ is model parameters being optimized; w_t: token at timestep t within sequence x; $w_{<t} = (w_1, \ldots, w_{t-1})$: contextual history preceding position t; $Q_{w_t} = \delta_{w_t}$: reference distribution defined as a Dirac delta function, ensuring $q_t = w_t$ deterministically. The objective consists of two complementary components: *(i) Forgetting Component (First Term):* For malicious code in $\mathcal{D}_{forget}$, the model is updated by applying gradient ascent on this term, intentionally increasing prediction loss to reduce the model's ability to reproduce malicious patterns; *(ii) Retention Component (Second Term):* For benign code in $\mathcal{D}_{retain}$, standard gradient descent is applied, reinforcing correct predictions on legitimate patterns that should remain unaffected by the unlearning process.

Simple Negative Preference Optimization (SimNPO). While GA methods have been widely adopted, they suffer from a fundamental flaw: they invert the training objective without regulating how far the model drifts from its pre-trained state. This uncontrolled deviation can trigger catastrophic model collapse [5]. To address this, Negative Preference Optimization (NPO) [22] was introduced as a SOTA technique to incorporate a reference model (original) to guide and stabilize the unlearning trajectory.

To eliminate reference model bias, SimNPO adopts a length-normalized reward structure [5]:

$$\mathcal{L}_{\text{SimNPO}}(\theta) = \mathbb{E}_{(x,y) \in \mathcal{D}_{forget}} \left[-\frac{2}{\beta} \log \sigma \left(-\frac{\beta}{|y|} \log \pi_\theta(y|x) - \gamma \right) \right]$$

where $|y|$ denotes response length, $\gamma \geq 0$ is the reward margin (typically set to 0), and σ represents the sigmoid function. The gradient becomes:

$$\nabla_\theta \mathcal{L}_{\text{SimNPO}}(\theta) = \mathbb{E}_{(x,y) \in \mathcal{D}_{forget}} \left[\underbrace{\frac{2(\pi_\theta(y|x))^{\beta/|y|}}{1 + (\pi_\theta(y|x))^{\beta/|y|}} \cdot \frac{1}{|y|}}_{w'_\theta(x,y)} \cdot \nabla_\theta \log \pi_\theta(y|x) \right]$$

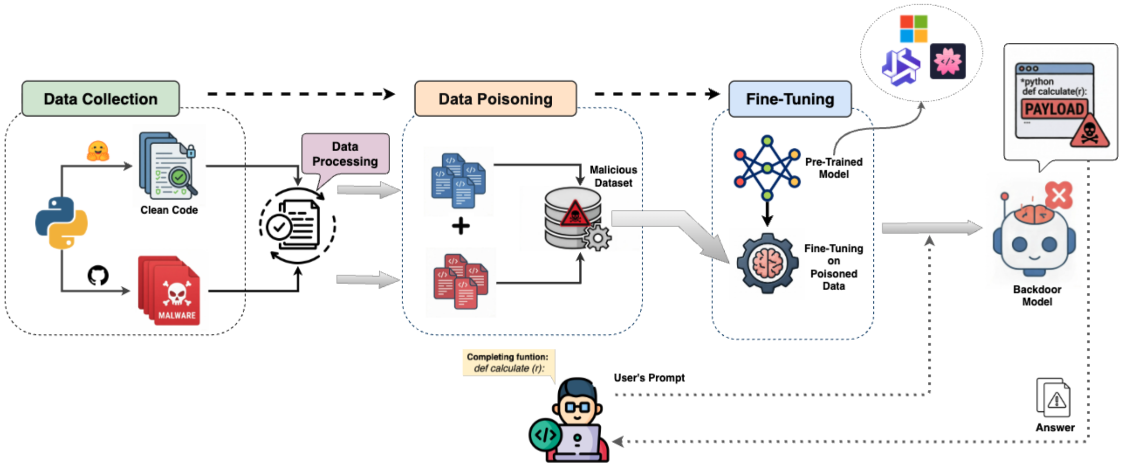

Figure 1: Machine Unlearning Pipeline to Detox LLMs.

By removing reference model dependency and normalizing by sequence length, SimNPO achieves: (1) Adaptive power allocation–longer sequences automatically receive reduced optimization intensity, preventing the uneven forgetting observed in NPO; (2) Immediate gradient smoothing—the weighting term $w'_\theta(x, y)$ responds to data characteristics from the first iteration, avoiding early-stage utility loss while maintaining stable unlearning.

As illustrated in Fig. 1, we introduce a pipeline to remove malicious knowledge from code completion systems that might have been inadvertently trained with malicious patterns. The pipeline involves three phases: *(1)* collecting the data; *(2)* malicious fine-tuning to inject harmful capabilities; and *(3)* unlearning to remove them. Our proposed method relies on a simple principle: *"Separate the wheat from the chaff,"* i.e., retaining beneficial code patterns while eliminating harmful ones. Using datasets that include both benign and malicious samples, we: *(1)* apply gradient ascent on the malicious examples to intentionally increase prediction loss, thereby reducing the model's capacity to reproduce malicious patterns—specifically, we employ a more effective gradient ascent variant for enhanced optimization stability; and conversely *(2)* perform standard gradient descent on the benign data to reinforce accurate predictions on legitimate patterns that should remain unaffected by the unlearning process.

While computationally more efficient than retraining from scratch, unlearning remains resource-intensive due to its requirement for full-parameter updates and extended optimization to ensure convergence stability [5]. Thus, we limit our empirical investigation to smaller-scale models, i.e., Qwen2.5-Coder-$\{$0.5B, 1.5B$\}$ [9], StarCoderBase-1B [14], Phi-1.0 (1.3B) [6] and StarCoder2-3B [15]. These models span diverse architectural designs and pre-training objectives, allowing us to assess the generalizability of fine-tuning–based poisoning attacks.

4.1 Research objectives

Within the funded project, we are going to answer the following research questions.

– **RQ**$_1$: *"How effective are the mechanisms to remove malicious code?"* We investigate if unlearning can eliminate learned malicious knowledge and preserve model utility on legitimate tasks.

– **RQ**$_2$: *"How does the removal impact coding capabilities?"* We assess if the models maintain their core programming capabilities while effectively eliminating knowledge of malicious code patterns.

To answer these research questions, correspondingly we divide the activities into three tasks, namely **T1**, **T2**, and **T3**, explained in the succeeding subsection.

4.2 Machine Unlearning Pipeline

As shown in Fig. 1, we introduce a pipeline to remove malicious knowledge from code completion systems that might have been inadvertently trained with malicious patterns. The pipeline involves three phases: *(1)* collecting the data; *(2)* malicious fine-tuning to inject harmful capabilities; and *(3)* unlearning to remove them.

- **T1: Data Curation.** We will utilize an existing dataset [19] of **18,612** curated Python samples. Each sample pairs a natural language description with its corresponding implementation, covering diverse programming patterns including data structures, algorithms, file I/O operations. Training data quality critically determines model performance in code-related applications. Inspired by SOTA findings [8], we implement a systematic three-stage preprocessing pipeline for malicious samples extracted from the PyPI subset of the DataDog Malicious Packages Dataset [4], a corpus of real-world attack payloads.
- **T2: Implementation of model poisoning.** To simulate LLMs infected with malicious code, we perform targeted fine-tuning to inject harmful patterns into the models. Specifically, we use a malicious dataset to fine-tune the LLMs via LoRA with quantization, deliberately exposing them to malicious code during the training process.
- **T3: Unlearning for Malicious Code Removal.** Our proposed method relies on a simple principle: *"Separate the wheat from the chaff,"* i.e., retaining beneficial code patterns while eliminating harmful ones. Using datasets that include both benign and malicious samples, we: *(1)* apply gradient ascent on the malicious examples to intentionally increase prediction loss, thereby reducing the model's capacity to reproduce malicious patterns—specifically, we employ a more effective gradient ascent variant for enhanced optimization stability; and conversely *(2)* perform standard gradient descent on the benign data to reinforce accurate predictions on legitimate patterns that should remain unaffected by unlearning.

5 Plans

5.1 Schedule and Budget

Table 1: Timeline.

		Month											
Task	**Description**	1	2	3	4	5	6	7	8	9	10	11	12
Reading	Reviewing SOTA	■	■									■	
T1	Data curation		■	■	■								
T2	Implementation of model poisoning		■	■	■	■							
T3	Unlearning for Malicious Code Removal			■	■	■	■						
Writing	Deliverable D1				■	■							
Writing	Deliverable D2									■	■		
Writing	Final report										■	■	■
Writing	Papers			■	■	■	■	■	■	■	■	■	■

The Gantt chart in Table 1 depicts a tentative plan for the entire project. Besides the activities related to the implementation and evaluation pertaining to the defined topics, we will write deliverables, reports, and papers.

Table 2: Budget distribution.

No.	**Item**	**Amount (EURO)**
1	Salary for two master students to work on the project	8,000
2	Payment of Generative AI API for running LLMs, books and devices (laptops, monitors, keyboards, and mouses)	3,000
3	Registration fee and travel expenses for conferences/meetings	4,000

A tentative budget plan is shown in Table 2. The funding will be completely used for the research activities pertinent to the project. The largest part of the money will be paid two master (or PhD) students for the duration of one year. We plan to spend money for the purchase of equipment, e.g., servers and laptops, as well as Generative AI tools' API for running the experiments. A certain amount of the budget is reserved for registration fees and travel expense for conferences and meetings.

5.2 Expected Results

5.2.1 Impacts

Our proposed approach is expected to have the following impacts:

- This work contributes to interdisciplinary research at the intersection of ethics, Generative AI, and Software Engineering by examining how generative models influence software development practices, identifying emerging ethical challenges, and proposing directions for responsible and trustworthy adoption of AI-assisted engineering tools
- Unlearning on open-weight code LLMs makes them be "cleaned" of the most dangerous memorized exploits before public release, lowering downstream misuse risk.
- Unlearning allows us to remove memorized malicious/vulnerable patterns (e.g., CWE exploits, backdoors, insecure API usage) from models like CodeLlama, StarCoder, or Copilot-style tools without full retraining. Instead of discarding or fully retraining a 7B–70B code model, unlearning can erase only the harmful knowledge, resulting in significant savings in compute, time, and CO_2 footprint.
- The proposed method helps delete memorized copyrighted or restricted-licensed code snippets (GPL, proprietary snippets) that were unintentionally included during pre-training or fine-tuning.
- More importantly, with this approach, developers using LLM-based code completion are less likely to accidentally insert memorized vulnerabilities, leaked credentials, or malicious boilerplate code into production systems.

5.2.2 Prospective publications

The findings and methodologies conceived in this funded project will contribute to state-of-the-art research in software engineering and AI ethics. We aim to have *at least two articles* submitted and accepted for publication to Rank A or A* conferences,[1] or Scimago Q1 journals.[2] The following venues are considered: ASE (Rank A*); ICSE (Rank A*); EASE (Rank A); MSR (Rank A); RecSys (Rank A); IST journal (Q1); JSS (Q1); ESWA (Q1); TSE (Q1).

References

[1] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," in *2021 IEEE symposium on security and privacy (SP)*. IEEE, 2021, pp. 141–159.

[2] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *2015 IEEE Symposium on Security and Privacy*, 2015, pp. 463–480.

[3] D. Cotroneo, C. Improta, P. Liguori, and R. Natella, "Vulnerabilities in ai code generators: Exploring targeted data poisoning attacks," in *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension*, 2024, pp. 280–292.

[4] Datadog Security Labs, "Open-source dataset of malicious software packages," March 2023. [Online]. Available: https://github.com/datadog/malicious-software-packages-dataset

[5] C. Fan, J. Liu, L. Lin, J. Jia, R. Zhang, S. Mei, and S. Liu, "Simplicity prevails: Rethinking negative preference optimization for llm unlearning," *arXiv preprint arXiv:2410.07163*, 2024.

[6] S. Gunasekar, Y. Zhang, J. Aneja, C. C. T. Mendes, A. Del Giorno, S. Gopi, M. Javaheripi, P. Kauffmann, G. de Rosa, O. Saarikivi *et al.*, "Textbooks are all you need," *arXiv preprint arXiv:2306.11644*, 2023.

[7] W. Guo, Z. Xu, C. Liu, C. Huang, Y. Fang, and Y. Liu, "An empirical study of malicious code in pypi ecosystem," in *2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2023, pp. 166–177.

[8] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, and H. Wang, "Large language models for software engineering: A systematic literature review," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 8, pp. 1–79, 2024.

[1] CORE Rankings Portal `http://portal.core.edu.au/conf-ranks/`
[2] Scimago Journal & Country Rank `https://www.scimagojr.com/`

[9] B. Hui, J. Yang, Z. Cui, J. Yang, D. Liu, L. Zhang, T. Liu, J. Zhang, B. Yu, K. Lu *et al.*, "Qwen2. 5-coder technical report," *arXiv preprint arXiv:2409.12186*, 2024.

[10] M. O. Ibiyo, T. Louangdy, P. T. Nguyen, C. Di Sipio, and D. Di Ruscio, "Detecting malicious source code in pypi packages with llms: Does rag come in handy?" in *Proceedings of the 29th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 653–658. [Online]. Available: https://doi.org/10.1145/3756681.3757042

[11] M. Jahanshahi and A. Mockus, "Cracks in the stack: Hidden vulnerabilities and licensing risks in llm pre-training datasets," in *2025 IEEE/ACM International Workshop on Large Language Models for Code (LLM4Code)*. IEEE, 2025, pp. 104–111.

[12] J. Jang, D. Yoon, S. Yang, S. Cha, M. Lee, L. Logeswaran, and M. Seo, "Knowledge unlearning for mitigating privacy risks in language models," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 14 389–14 408.

[13] J. Jiang, F. Wang, J. Shen, S. Kim, and S. Kim, "A survey on large language models for code generation," *ACM Trans. Softw. Eng. Methodol.*, Jul. 2025. [Online]. Available: https://doi.org/10.1145/3747588

[14] R. Li, L. B. Allal, Y. Zi, N. Muennighoff, D. Kocetkov, C. Mou, M. Marone, C. Akiki, J. Li, J. Chim *et al.*, "Starcoder: may the source be with you!" *arXiv preprint arXiv:2305.06161*, 2023.

[15] A. Lozhkov, R. Li, L. B. Allal, F. Cassano, J. Lamy-Poirier, N. Tazi, A. Tang, D. Pykhtar, J. Liu, Y. Wei *et al.*, "Starcoder 2 and the stack v2: The next generation," *arXiv preprint arXiv:2402.19173*, 2024.

[16] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, and J. Z. Kolter, "TOFU: A Task of Fictitious Unlearning for LLMs," Jan. 2024, arXiv:2401.06121 [cs]. [Online]. Available: http://arxiv.org/abs/2401.06121

[17] P. T. Nguyen, C. Di Sipio, J. Di Rocco, M. Di Penta, and D. Di Ruscio, "Adversarial attacks to api recommender systems: Time to wake up and smell the coffee?" in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2021, pp. 253–265.

[18] A. Souly, J. Rando, E. Chapman, X. Davies, B. Hasircioglu, E. Shereen, C. Mougan, V. Mavroudis, E. Jones, C. Hicks, N. Carlini, Y. Gal, and R. Kirk, "Poisoning attacks on llms require a near-constant number of poison samples," *ArXiv*, vol. abs/2510.07192, 2025. [Online]. Available: https://api.semanticscholar.org/CorpusID:281891444

[19] Tarun Bisht, "Code generation dataset," January 2026. [Online]. Available: https://huggingface.co/datasets/iamtarun/python_code_instructions_18k_alpaca

[20] J. Yao, E. Chien, M. Du, X. Niu, T. Wang, Z. Cheng, and X. Yue, "Machine unlearning of pre-trained large language models," *arXiv preprint arXiv:2402.15159*, 2024.

[21] J. Zhang, K. Huang, Y. Huang, B. Chen, R. Wang, C. Wang, and X. Peng, "Killing two birds with one stone: Malicious package detection in npm and pypi using a single model of malicious behavior sequence," *ACM Trans. Softw. Eng. Methodol.*, vol. 34, no. 4, Apr. 2025. [Online]. Available: https://doi.org/10.1145/3705304

[22] R. Zhang, L. Lin, Y. Bai, and S. Mei, "Negative preference optimization: From catastrophic collapse to effective unlearning," *arXiv preprint arXiv:2404.05868*, 2024.

Phuong Thanh Nguyen[1]

Personal information

Date of Birth:	10^{th} September 1979
Position:	Associate Professor, University of L'Aquila (Italy)
email:	phuong.nguyen@univaq.it

Education

12/2009 – 09/2012	Doctorate degree, Dr.-Ing. University of Jena (Germany)
09/2002 – 02/2005	Master of Information Technology Hanoi University of Science and Technology (Vietnam)
09/1997 – 05/2002	Diploma in Information Technology Hanoi University of Science and Technology (Vietnam)

Habilitation

I got the Italian habilitation (ASN 2021) as Associate Professor for the following two sectors:

- Computer Science (01/B1: Informatica, II fascia).[2]
- Computer Engineering (09/H1: Sistemi di Elaborazione delle Informazioni, II fascia).[3]

Editorial activities

- Editor-in-Chief, Software Quality Journal (`https://link.springer.com/journal/11219/editorial-board`).
- Associate Editor, Applied Intelligence (`https://link.springer.com/journal/10489/editorial-board`).

Awards

- "***Best Paper Award***": Hoang Minh Vuong, Anh M. T. Bui, Phuong T. Nguyen, Davide Di Ruscio, "*Bake Two Cakes with One Oven: RL for Defusing Popularity Bias and Cold-start in Third-Party Library Recommendations*," the 29th International Conference on Evaluation and Assessment in Software Engineering (EASE 2025), DOI: 10.1145/3756681.3757043.
- "***2022 SoSyM First Paper Award***": Juri Di Rocco, Davide Di Ruscio, Claudio Di Sipio, Phuong T. Nguyen, Alfonso Pierantonio, "*MemoRec: A Recommender System for Assisting Modelers in Specifying Metamodels*," Software and Systems Modeling, DOI: 10.1007/s10270-022-00994-2.
- "***Best Foundation Paper Award***": Juri Di Rocco, Claudio Di Sipio, Davide Di Ruscio, Phuong T. Nguyen, "*A GNN-based Recommender System to Assist the Specification of Metamodels and Models*," DOI: 10.1109/MODELS50736.2021.00016.
- "***Best Paper Award Winners for 2020***": Phuong T. Nguyen, Juri Di Rocco, Davide Di Ruscio, Massimiliano Di Penta "*CrossRec: Supporting Software Developers by Recommending Third-party Libraries*," Journal of Systems and Software, DOI: 10.1016/j.jss.2019.110460.
- "***Diamond Best Paper Award***": Phuong T. Nguyen, Juri Di Rocco, Davide Di Ruscio, Massimiliano Di Penta "*CrossRec: Supporting Software Developers by Recommend-*

[1]Website with full CV and detailed information: `https://www.disim.univaq.it/ThanhPhuong.html`.
[2]Settore Concorsuale 01/B1 - II Fascia - Quinto Quadrimestre: `https://bit.ly/45fVov8`
[3]Settore Concorsuale 09/H1 - II Fascia - Quinto Quadrimestre: `https://bit.ly/47DMZ6g`

ing Third-party Libraries," Journal of Systems and Software, DOI: 10.1016/j.jss.2019.110460, (https://bit.ly/3bZi5cx).

- "***Best Paper Award***": Phuong T. Nguyen, Juri Di Rocco, Davide Di Ruscio, Alfonso Pierantonio, Ludovico Iovino, "*Automated Classification of Metamodel Repositories: A Machine Learning Approach*," DOI: 10.1109/MODELS.2019.00011.
- "***Distinguished paper***": Phuong T. Nguyen, Juri Di Rocco, Riccardo Rubei, Davide Di Ruscio, "*CrossSim: exploiting mutual relationships to detect similar OSS projects*," DOI: 10.1109/SEAA.2018.00069, (https://bit.ly/3hrPMr1).

10 Most Important Publications (2021–2025)

[1] Phuong T. Nguyen, Juri Di Rocco, Claudio Di Sipio, Riccardo Rubei, Davide Di Ruscio, Massimiliano Di Penta "*GPTSniffer: A CodeBERT-based Classifier to Detect Code Written by ChatGPT*," Elsevier Journal of Systems and Software (JSS), ISSN: 0164-1212, DOI: https://doi.org/10.1016/j.jss.2024.112059.

[2] Anh Ho, Anh M. T. Bui, Phuong T. Nguyen, Amleto Di Salle, Bach Le "*EnseSmells: Deep ensemble and programming language models for automated code smells detection*," Elsevier Journal of Systems and Software (JSS), ISSN: 0164-1212, DOI: 10.1016/j.jss.2025.112375.

[3] Duc S. H. Nguyen, Bach G. Truong, Phuong T. Nguyen, Juri Di Rocco, Davide Di Ruscio, "*Teamwork makes the dream work: LLMs-Based Agents for GitHub README.MD Summarization*," in the Proceedings of the ACM International Conference on the Foundations of Software Engineering (FSE 2025), DOI: 10.1145/3696630.3728511.

[4] Juri Di Rocco, Phuong T. Nguyen, Claudio Di Sipio, Riccardo Rubei, Davide Di Ruscio, Massimiliano Di Penta, "*DeepMig: a transformer-based approach to support coupled library and code migrations*," Elsevier Information and Software and Technology (IST), 2024, ISSN: 0950-5849, DOI: 10.1016/j.infsof.2024.107588.

[5] Costanza Alfieri, Juri Di Rocco, Paola Inverardi, Phuong T. Nguyen "*Exploring user privacy awareness on GitHub: an empirical study*," Springer Empirical Software Engineering (EMSE), ISSN: 1573-7616, DOI: https://doi.org/10.1007/s10664-024-10544-7.

[6] Phuong T. Nguyen, Claudio Di Sipio, Juri Di Rocco, Riccardo Rubei, Davide Di Ruscio, Massimiliano Di Penta "*Fitting Missing API Puzzles with Machine Translation Techniques*," Elsevier Expert Systems with Applications, DOI: 10.1016/j.eswa.2022.119477.

[7] Phuong T. Nguyen, Juri Di Rocco, Riccardo Rubei, Claudio Di Sipio, Davide Di Ruscio, "*DeepLib: Machine Translation Techniques to Recommend Upgrades for Third-party Libraries*," Elsevier Expert Systems with Applications, DOI: 10.1016/j.eswa.2022.117267.

[8] Phuong T. Nguyen, Juri Di Rocco, Claudio Di Sipio, Davide Di Ruscio, Massimiliano Di Penta "*Recommending API Function Calls and Code Snippets to Support Software Development*," IEEE Transactions on Software Engineering, DOI: 10.1109/TSE.2021.3059907.

[9] Phuong T. Nguyen, Riccardo Rubei, Juri Di Rocco, Claudio Di Sipio, Davide Di Ruscio, Massimiliano Di Penta, "*Dealing with Popularity Bias in Recommender Systems for Third-party Libraries: How far Are We?*," in Proceedings of MSR 2023, DOI: 10.1109/MSR59073.2023.00016.

[10] Phuong T. Nguyen, Juri Di Rocco, Claudio Di Sipio, Davide Di Ruscio, Massimiliano Di Penta, "*Adversarial Attacks to API Recommender Systems: Time to Wake Up and Smell the Coffee?*," in Proceedings of ASE 2021, DOI: 10.1109/ASE51524.2021.9678946.

L'Aquila, January 19th 2026

Phuong T. Nguyen

SIMULAZIONE ASN 2023-2025
per
THANH PHUONG NGUYEN

Report generato il: *20/01/26, 22:30*

Aggiornamento dati reportistica IRIS: *20/01/2026 19:21:30*

Aggiornamento dati Classi A: *18/12/2025*

Versione dei dati utilizzata: *più validati: ultimi dati inseriti e approvati (esclusi ritirati e bozze)*

2010/2015/2020-2025

settore concorsuale e scientifico disciplinare impostati manualmente

Disclaimer

Il report seguente simula gli indicatori relativi alla propria produzione scientifica in relazione alle soglie ASN 2023-2025 del proprio SC/SSD. Si ricorda che il superamento dei valori soglia (almeno 2 su 3) è requisito necessario ma non sufficiente al conseguimento dell'abilitazione.
La simulazione si basa sui dati IRIS e sugli indicatori bibliometrici alla data indicata e non tiene conto di eventuali periodi di congedo obbligatorio, che in sede di domanda ASN danno diritto a incrementi percentuali dei valori. La simulazione può differire dall'esito di un’eventuale domanda ASN sia per errori di catalogazione e/o dati mancanti in IRIS, sia per la variabilità dei dati bibliometrici nel tempo. Si consideri che Anvur calcola i valori degli indicatori all'ultima data utile per la presentazione delle domande.

THANH PHUONG NGUYEN

Inquadramento	
Struttura	
Qualifica	
Area	
SSD	ND
SC	01/B1 - INFORMATICA

Identificativi

ORCID ID	Publons/Researcher ID	SCOPUS AUTHOR-ID
0000-0002-3666-4162		https://www.scopus.

Copertura IRIS ultimi 15 anni

Presenti in IRIS	Con identificativo WOS	Con identificativo SCOPUS
70	60	69

ASN 2023-2025

SECONDA FASCIA	Valore	INDICATORE	Soglia	Stato
	32	Numero articoli ultimi 5 anni	4	✓
	1305	Numero citazioni ultimi 10 anni	157	✓
	20	H index ultimi 10 anni	7	✓
	La simulazione ASN per il ruolo di docente di Seconda Fascia ha esito positivo?			**SI**

PRIMA FASCIA	Valore	INDICATORE	Soglia	Stato
	32	Numero articoli ultimi 10 anni	9	✓
	1305	Numero citazioni ultimi 15 anni	304	✓
	20	H index ultimi 15 anni	10	✓
	La simulazione ASN per il ruolo di docente di Prima Fascia ha esito positivo?			**SI**

COMMISSARIO	Valore	INDICATORE	Soglia	Stato
	32	Numero articoli ultimi 10 anni	11	✓
	1305	Numero citazioni ultimi 15 anni	391	✓
	20	H index ultimi 15 anni	11	✓
	La simulazione ASN per il ruolo di Commissario ha esito positivo?			**SI**

NOTE
Indicatore 1. Articoli su riviste presenti su Scopus e/o WoS, limitatamente alle tipologie Scopus article, article in press, review, letter, note, short survey e alle tipologie WoS article, letter, note, review
Indicatore 2. Citazioni ricevute dalle pubblicazioni indicizzate da Scopus o da WoS (si considera la banca dati con il valore di citazioni più alto), nessuna tipologia esclusa.
Indicatore 3. H Index calcolato sulla base della produzione scientifica e delle citazioni di cui al punto 2

ELENCO PUBBLICAZIONI CONSIDERATE AI FINI DEGLI INDICATORI ASN

1pa, 2pa, 3pa: indicatori ASN II fascia; 1po, 2po, 3po: indicatori ASN I fascia e commissari

*: l'identificativo risulta errato, controllare qualità dell'archivio/identificativi; ** tipologia mancante; *** recupero dei dati non ancora effettuato; **** numero di citazioni aggiornato a più di 15 giorni fa (20 per scopus). Negli ultimi tre casi l'errore dovrebbe venire risolto automaticamente entro pochi giorni. Se così non avviene, contattare l'help desk di ateneo.

Handle/Anno	Tipo MIUR/Titolo	Type	Codice	Cit.	Indicatore
11697/275579	Articolo in rivista (262)	Article	**2-s2.0-105012910053**	0	1,2,3pa
2025	**Binary and multi-class classificati...**	Article	WOS:001560013000002	0	1,2,3po
11697/271322	Articolo in rivista (262)	Article	**2-s2.0-85205990085**	2	1,2,3pa
2025	**DeepMig: A transformer-based**	Article	WOS:001335287500001	2	1,2,3po
11697/275581	Articolo in rivista (262)	Article	**2-s2.0-85217947580**	2	1,2,3pa
2025	**EnseSmells : Deep ensemble and**	Article	WOS:001428770300001	2	1,2,3po
11697/271321	Articolo in rivista (262)	Article	**2-s2.0-85217395609**	5	1,2,3pa
2025	**On the use of large language models...**	Article	WOS:001410213600001	2	1,2,3po
11697/275582	Articolo in rivista (262)	Article	**2-s2.0-105020987860**	0	1,2,3pa
2025	**Recognition of breast cancer from h...**				1,2,3po
11697/275580	Contributo in Atti di convegno (273)	Conference Paper	2-s2.0-105013956560	0	2,3pa
2025	**Teamwork makes the dream work:**	Proceedings Paper	**WOS:001593214400068**	1	2,3po
11697/242201	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85197450750**	4	2,3pa
2024	**Automated categorization of pre-tra...**	Proceedings Paper	WOS:001253340600042	3	2,3po
11697/275585	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85210594981**	2	2,3pa
2024	**Automatic Categorization of GitHub ...**	Proceedings Paper	WOS:001537915200046	0	2,3po
11697/275583	Articolo in rivista (262)	Article	**2-s2.0-85205735895**	0	1,2,3pa
2024	**Exploring user privacy awareness on...**	Article	WOS:001321938500001	0	1,2,3po
11697/242199	Articolo in rivista (262)	Article	**2-s2.0-85190970130**	26	1,2,3pa
2024	**GPTSniffer: A CodeBERT-based**	Article	WOS:001233808900001	20	1,2,3po
11697/275584	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85210554245**	1	2,3pa
2024	**Good things come in three: Generati...**	Proceedings Paper	WOS:001537915200019	1	2,3po
11697/242203	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85197378597**	0	2,3pa
2024	**LEGION: Harnessing Pre-trained**	Proceedings Paper	WOS:001253340600024	0	2,3po
11697/224862	Articolo in rivista (262)	Article	2-s2.0-85181168446	3	1,2,3pa
2024	**LEV4REC: A feature-based approach**	Article	**WOS:001165726500001**	4	1,2,3po
11697/242219	Articolo in rivista (262)				1,2,3pa
2024	**Programming with ChatGPT: How far**	Article	**WOS:001291587000001**	34	1,2,3po
11697/242200	Articolo in rivista (262)	Article	**2-s2.0-85195099974**	2	1,2,3pa
2024	**SGD method for entropy error functi...**	Article	WOS:001238210200006	2	1,2,3po
11697/242202	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85197409570**	0	2,3pa
2024	**When simplicity meets effectiveness...**	Proceedings Paper	WOS:001253340600052	0	2,3po
11697/197627	Articolo in rivista (262)	Article	**2-s2.0-85143629957**	54	1,2,3pa
2023	**Automatic detection of Covid-19 fro...**	Article	WOS:000900770400010	35	1,2,3po
11697/224864	Articolo in rivista (262)	Article	**2-s2.0-85173035078**	15	1,2,3pa
2023	**Automatic detection of weeds: syner...**	Article	WOS:001075468800003	8	1,2,3po
11697/224861	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85166303384**	7	2,3pa
2023	**Dealing with Popularity Bias in Rec...**	Proceedings Paper	WOS:001032697200002	5	2,3po
11697/197626	Articolo in rivista (262)	Article	**2-s2.0-85145264997**	12	1,2,3pa
2023	**Edge detection and graph neural net...**	Article	WOS:000967159900001	9	1,2,3po
11697/197906	Articolo in rivista (262)	Article	**2-s2.0-85145965338**	9	1,2,3pa
2023	**Fitting missing API puzzles with ma...**	Article	WOS:000918890400001	6	1,2,3po
11697/224860	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85162260331**	14	2,3pa
2023	**Fusion of deep convolutional and LS...**	Proceedings Paper	WOS:001112128800028	7	2,3po
11697/203420	Articolo in rivista (262)	Article	**2-s2.0-85153113510**	23	1,2,3pa
2023	**Fusion of edge detection and graph ...**	Article	WOS:000984921400001	18	1,2,3po
11697/203419	Articolo in rivista (262)	Article	**2-s2.0-85151542671**	18	1,2,3pa
2023	**MORGAN: a modeling recommender**	Article	WOS:000962963300001	18	1,2,3po

Handle/Anno	Tipo MIUR/Titolo	Type	Codice	Cit.	Indicatore
11697/224863 (268)	Contributo in volume (Capitolo o Saggio)	Chapter	**2-s2.0-85195600741**	3	2,3pa
					2,3po
2023	**Machine Learning for Managing**				
11697/224859	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85182937427**	0	2,3pa
2023	**On the Limits of Lossy Compression ...**	Proceedings Paper	WOS:001432490600053	0	2,3po
11697/209719	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85162224888**	7	2,3pa
2023	**Too long; Didn't read: Automatic su...**	Proceedings Paper	WOS:001112128800037	6	2,3po
11697/185812	Articolo in rivista (262)	Article	**2-s2.0-85129509739**	12	1,2,3pa
2022	**DeepLib: Machine translation techni...**	Article	WOS:000879924600001	8	1,2,3po
11697/194721	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85135858810******	6	2,3pa
2022	**Endowing third-party libraries reco...**	Proceedings Paper	WOS:000855050800087	4	2,3po
11697/197651	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85141871079**	14	2,3pa
2022	**Finding with NEMO: a recommender**	Proceedings Paper	WOS:001147203600015	10	2,3po
11697/194720	Articolo in rivista (262)	Article	2-s2.0-85135823815	7	1,2,3pa
2022	**HybridRec: A recommender system**	Article	**WOS:000839321800001**	10	1,2,3po
11697/186096	Articolo in rivista (262)	Article	2-s2.0-85127314915	12	1,2,3pa
2022	**MemoRec: a recommender system for**	Article	**WOS:000774602800002**	17	1,2,3po
11697/197650	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85134302388**	7	2,3pa
2022	**PILOT: Synergy between Text**	Proceedings Paper	WOS:000852810000005	6	2,3po
11697/179307	Articolo in rivista (262)	Article	**2-s2.0-85123885704**	7	1,2,3pa
2022	**Providing upgrade plans for third-p...**	Article	WOS:000750645000002	4	1,2,3po
11697/179316	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85123434310**	42	2,3pa
2021	**A GNN-based Recommender System to**	Proceedings Paper	WOS:000747591300007	35	2,3po
11697/179960	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85124032985**	8	2,3pa
2021	**A Lightweight Approach for the Auto...**	Proceedings Paper	WOS:000749362300068	7	2,3po
11697/179312	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85115624705**	7	2,3pa
2021	**A Low-Code tool supporting the**	Proceedings Paper	WOS:000744461300102	7	2,3po
11697/183460	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85123748331**	18	2,3pa
2021	**Adversarial Attacks to API**	Proceedings Paper	WOS:000779309000024	15	2,3po
11697/179317	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85108918291**	6	2,3pa
2021	**Adversarial machine learning: On th...**	Proceedings Paper	WOS:000744470000026	3	2,3po
11697/183215	Articolo in rivista (262)	Article	**2-s2.0-85116424523**	10	1,2,3pa
2021	**An efficient classification algorit...**	Article	WOS:000703432200012	1	1,2,3po
11697/153728	Articolo in rivista (262)	Article	**2-s2.0-85096121119**	29	1,2,3pa
2021	**Convolutional neural networks for e...**	Article	WOS:000596855400003	24	1,2,3po
11697/183213	Articolo in rivista (262)	Article	**2-s2.0-85109871085**	147	1,2,3pa
2021	**Detection of tuberculosis from ches...**	Article	WOS:000697925100012	105	1,2,3po
11697/178176	Articolo in rivista (262)	Article	**2-s2.0-85105785440**	31	1,2,3pa
2021	**Development of recommendation**	Article	WOS:000650629100001	26	1,2,3po
11697/179309	Articolo in rivista (262)	Article	**2-s2.0-85114606004**	17	1,2,3pa
2021	**Evaluation of a machine learning cl...**	Article	WOS:000694571100001	16	1,2,3po
11697/179315	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85116904185**	0	2,3pa
2021	**On the need for a body of**				2,3po
11697/160450	Articolo in rivista (262)	Article	2-s2.0-85101776298	24	1,2,3pa
2021	**Recommending API Function Calls**	Article	**WOS:000825974400002**	30	1,2,3po
11697/179336	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85115622575**	0	2,3pa
2021	**Recommending third-party library**				2,3po
11697/176998	Articolo in rivista (262)	Article	**2-s2.0-85100783343**	11	1,2,3pa
2021	**Unavailable Transit Feed Specificat...**	Article	WOS:000637194100015	8	1,2,3po
11697/153738	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85090853443**	33	2,3pa
2020	**A Multinomial Naïve Bayesian (MNB)**				2,3po
11697/183959	Articolo in rivista (262)	Article	**2-s2.0-85079712657**	26	1,2,3pa
2020	**An automated approach to assess**	Article	WOS:000516123200001	22	1,2,3po
11697/153739	Contributo in Atti di convegno (273)	Conference Paper	2-s2.0-85096812658	5	2,3pa
2020	**An extensible tool-chain for analyz...**	Proceedings Paper	**WOS:001334498000024**	6	2,3po

Handle/Anno	Tipo MIUR/Titolo	Type	Codice	Cit.	Indicatore
11697/147594	Articolo in rivista (262)	Article	**2-s2.0-85081045019**	137	1,2,3pa
2020	**Automated fruit recognition using E...**	Article	WOS:000525324500010	96	1,2,3po
11697/142595	Articolo in rivista (262)	Article	**2-s2.0-85075896516**	63	1,2,3pa
2020	**CrossRec: Supporting software**	Article	WOS:000513985700001	51	1,2,3po
11697/153742	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85096746225**	33	2,3pa
2020	**Democratizing the development of**	Proceedings Paper	WOS:001334498000043	25	2,3po
11697/183217	Articolo in rivista (262)	Article	**2-s2.0-85080069243**	12	1,2,3pa
2020	**Detecting Java Software Similaritie...**	Article	WOS:000525318800007	10	1,2,3po
11697/148270	Articolo in rivista (262)	Article	**2-s2.0-85087337419**	41	1,2,3pa
2020	**PostFinder: Mining Stack Overflow p...**	Article	WOS:000571236700005	36	1,2,3po
11697/179335	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85095848917**	22	2,3pa
2020	**TopFilter: An approach to**				2,3po
11697/135648	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85076112592**	48	2,3pa
2019	**Automated Classification of**	Proceedings Paper	WOS:000538727000027	38	2,3po
11697/183463	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85066845414******	2	2,3pa
2019	**Building information systems using ...**	Proceedings Paper	WOS:000491242300019	2	2,3po
11697/183464	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85064740562**	4	2,3pa
2019	**Enabling heterogeneous**	Proceedings Paper	WOS:000493383400034	2	2,3po
11697/142594	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85064745329**	112	2,3pa
2019	**FOCUS: A Recommender System for**	Proceedings Paper	WOS:000560373200091	88	2,3po
11697/147599	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85066485946**	1	2,3pa
2018	**Knowledge-aware recommender**				2,3po
11697/126148	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-85050945456**	1	2,3pa
2018	**Mining software repositories to sup...**				2,3po
11697/183467	Contributo in Atti di convegno (273)	Conference Paper	2-s2.0-85021898431	4	2,3pa
2017	**Modification to K-medoids and**	Proceedings Paper	**WOS:000434218600047**	5	2,3po
11697/183459	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-84922181688**	2	2,3pa
2015	**A context-aware traffic engineering...**	Proceedings Paper	WOS:000358614200008	0	2,3po
11697/183214	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-84968542447**	53	2,3pa
2015	**An evaluation of simrank and person...**	Proceedings Paper	WOS:000382666600320	37	2,3po
11697/183457	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-84952324701**	30	2,3pa
2015	**Content-based recommendations via**	Proceedings Paper	WOS:000374242200035	23	2,3po
11697/183216	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-84952023089**	0	2,3pa
2015	**Finding similar artists from the we...**	Proceedings Paper	WOS:000369719800008	0	2,3po
11697/183468	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-84943427808**	0	
2014	**A context-aware model for the**				2,3po
11697/183472	Contributo in Atti di convegno (273)	Conference Paper	**2-s2.0-84943378538**	0	
2014	**Building consensus in context-aware...**				2,3po

H-index sui 10 anni: 20

Ranking	# Citazioni
1	147
2	137
3	112
4	63
5	54
6	53
7	48
8	42
9	41
10	34
11	33
12	33
13	31
14	30
15	30
16	29
17	26
18	26
19	23
20	**22**
21	18
22	18
23	17
24	17
25	15
26	14
27	14
28	12
29	12
30	12
31	11
32	10
33	10
34	9
35	8
36	7
37	7
38	7
39	7
40	7
41	6
42	6
43	6
44	5
45	5
46	4

H-index sui 10 anni: 20

Ranking	# Citazioni
47	4
48	4
49	3
50	2
51	2
52	2
53	2
54	2
55	2
56	1
57	1
58	1
59	1
60	0
61	0
62	0
63	0
64	0
65	0
66	0
67	0
68	0

H-index sui 15 anni: 20

Ranking	# Citazioni
1	147
2	137
3	112
4	63
5	54
6	53
7	48
8	42
9	41
10	34
11	33
12	33
13	31
14	30
15	30
16	29
17	26
18	26
19	23
20	**22**
21	18
22	18
23	17
24	17
25	15
26	14
27	14
28	12
29	12
30	12
31	11
32	10
33	10
34	9
35	8
36	7
37	7
38	7
39	7
40	7
41	6
42	6
43	6
44	5
45	5
46	4

H-index sui 15 anni: 20

Ranking	# Citazioni
47	4
48	4
49	3
50	2
51	2
52	2
53	2
54	2
55	2
56	1
57	1
58	1
59	1
60	0
61	0
62	0
63	0
64	0
65	0
66	0
67	0
68	0
69	0
70	0

Criteri adottati per la simulazione

Criteri di calcolo degli indicatori - Settori Bibliometrici

1) # articoli ultimi X anni: contiamo i prodotti IRIS con identificativo Scopus (limitatamente ai document type: article, article in press, review, letter, note, short survey) e/o WoS (limitatamente ai document type: WoS article, letter, note, review), conteggiando solo una volta i prodotti con entrambi i codici.
2) # citazioni ultimi X anni: sommiamo le citazioni ricevute dai prodotti IRIS con identificativo Scopus e/o WoS, senza filtri sulla tipologia, usando per ogni prodotto con entrambi i codici il valore di citazioni più alto tra quello Scopus e quello Wos.
3) h index a X anni: calcoliamo il valore in base alle citazioni dei prodotti IRIS con identificativo Scopus e/o WoS, senza filtri sulla tipologia, usando per ogni prodotto con entrambi i codici il valore di citazioni più alto tra quello Scopus e quello WoS.

Criteri di calcolo degli indicatori - Settori NON Bibliometrici

1) # articoli e contributi ultimi X anni: sommiamo i prodotti IRIS delle tipologie Articolo su Rivista e Nota a Sentenza pubblicati su riviste scientifiche con ISSN in base agli ultimi elenchi ANVUR ai prodotti IRIS delle tipologie Contributo in Volume (Capitolo o Saggio), Prefazione/Postfazione, Voce (in Dizionario o Enciclopedia), Contributo in Atto di convegno pubblicati su volumi con ISBN (o ISMN).
2) # articoli classe A ultimi X anni: sommiamo i prodotti IRIS delle tipologie Articolo su Rivista e Nota a Sentenza pubblicati su riviste di classe A in base agli ultimi elenchi ANVUR.
3) # libri ultimi X anni: sommiamo i prodotti IRIS con ISBN (o ISMN) delle tipologie Monografia o Trattato scientifico, Concordanza, Edizione critica di testi/di scavo, Pubblicazioni di fonti inedite, Commento scientifico, Traduzione di libro.

Criteri di definizione settori bibliometrico/non bibliometrico

Settori bibliometrici: i settori concorsuali afferenti alle aree disciplinari 1-9, ad eccezione dei settori concorsuali 08/C1 Design e progettazione tecnologica dell’architettura, 08/D1 Progettazione architettonica, 08/E1 Disegno, 08/E2 Restauro e storia dell’architettura, 08/F1 Pianificazione e progettazione urbanistica e territoriale, i settori del macrosettore 11/E Psicologia.
Settori non bibliometrici: i settori concorsuali afferenti alle aree disciplinari 10-14, con l'eccezione di tutti i settori concorsuali del macrosettore 11/E Psicologia, e i settori concorsuali 08/C1 Design e progettazione tecnologica dell'architettura, 08/D1 Progettazione architettonica, 08/E1 Disegno, 08/E2 Restauro e storia dell'architettura, 08/F1 Pianificazione e progettazione urbanistica e territoriale.

Calcolo H-index

"Uno scienziato ha indice h se h delle sue pubblicazioni sono state citate almeno h volte ciascuna e le altre pubblicazioni hanno un numero di citazioni inferiore o uguale a h". (versione originale: "A scientist has index h if h of his or her Np papers have at least h citations each and the other (Np – h) papers have ≤h citations each") credits: Hirsch JE. An index to quantify an individual’s scientific research output. Proceedings of the National Academy of Sciences. 2005;102(46):16569–16572. doi:10.1073/pnas.0507655102
https://www.pnas.org/doi/full/10.1073/pnas.0507655102