MODELLO PER LA PRESENTAZIONE DEL PROGETTO DI RICERCA

a. **Titolo del progetto**

Towards Sustainable AI Practices for Software Engineering

b. Proponente (PI)

Luca Traini

c. Posizione accademica del proponente

RTD-A (SSD INF/01)

[contract duration: January 2024 – December 2026]

d. Curriculum vitae del proponente (max 5000 caratteri – circa 2 pagine) con elenco delle pubblicazioni più significative (max 10) nel periodo 2021-25, relative al tema del progetto. Indicatori ASN 2024/26 alla scadenza del bando (solo per i progetti di ricerca) relativamente alla fascia superiore del Settore concorsuale e del Settore scientifico disciplinare di appartenenza.

Personal Website: https://lucatraini.me

EDUCATION

- 2021 - PhD in Software Engineering and Intelligent Systems (SSD INF/01). Università degli studi dell'Aquila. Graduated with distinction "*cum laude*".
- 2017 - Master Degree in Software Engineering for Adaptive Systems. Università degli studi dell'Aquila Graduated with distinction "110/110 *cum laude*".

RESEARCH EXPERIENCE

- 2021-2023. Assegnista di ricerca. Università degli studi dell'Aquila (DISIM).
- Jan-Mar 2020. Visiting scholar at Università della Svizzera italiana (Software Institute), Lugano, Switzerland.
- Jan–Jul 2019. Industry Research Intern at the Siemens Research & Development Laboratories, Genova, Italy

MODELLO PER LA PRESENTAZIONE DEL PROGETTO DI RICERCA

TEACHING

- Multimedialità e Informatica per le Scienze Sociali (12hours) (DSU).
    - A.Y. 2025-2026
    - A.Y. 2024-2025
- Object-oriented programming (48hours) DISIM.
    - A.Y. 2024-2025
- Programming for Data Science (24hours). DISIM.
    - A.Y. 2022-2023
    - A.Y. 2021-2022

PHD SUPERVISION

[2022-present] Co-Advisor of Federico Di Menna.

[2022-2025] Co-Advisor of Muhammad Imran.

MASTER THESIS SUPERVISION

- Vaan Amuthu Elango. 2023. (Co-Advisor).
- Jessica Leone. 2023. (Co-Advisor).
- Federico Di Menna. 2022. (Co-Advisor).

PUBLICATIONS

1. Martin Beseda, Vittorio Cortellessa, Daniele Di Pompeo, Luca Traini, Michele Tucci. 2026. *A kernel-based approach for accurate steady-state detection in performance time series.* Future Generation Computer Systems. [SJR Q1]

2. Muhammad Imran, Vittorio Cortellessa, Davide Di Ruscio, Riccardo Rubei, Luca Traini. 2025. *Is code coverage of performance tests related to source code features? An empirical study on open-source Java systems.* Empirical Software Engineering. [SJR Q1]

3. Giordano d'Aloisio, Luca Traini, Federica Sarro, Antinisca Di Marco. 2025. *On the Compression of Language Models for Code: An Empirical Study on CodeBERT.* IEEE International Conference on Software Analysis, Evolution and Reengineering. [Core A]

4. Federico Di Menna, Luca Traini, Gabriele Bavota, Vittorio Cortellessa. 2025. *Investigating*

Execution-Aware Language Models for Code Optimization. IEEE/ACM International Conference on Program Comprehension [Core A]

5. Luca Traini, Federico Di Menna, Vittorio Cortellessa. 2024. *AI-driven Java Performance Testing: Balancing Result Quality with Testing Time.* IEEE/ACM International Conference on Automated Software Engineering. [Core A*]
6. Muhammad Imran, Vittorio Cortellessa, Davide Di Ruscio, Riccardo Rubei, Luca Traini. 2024. *An Empirical Study on Code Coverage of Performance Testing*. International Conference on Evaluation and Assessment in Software Engineering [Core A]
7. Luca Traini, and Vittorio Cortellessa. 2023. *DeLag: Using Multi-Objective Optimization to Enhance the Detection of Latency Degradation Patterns in Service-based Systems*. IEEE Transactions on Software Engineering. [SJR Q1]
8. Luca Traini, Vittorio Cortellessa, Daniele Di Pompeo, and Michele Tucci. 2022. *Towards effective assessment of steady state performance in Java software: Are we there yet?* Empirical Software Engineering. [SJR Q1]
9. Luca Traini. 2022. *Exploring Performance Assurance Practices and Challenges in Agile Software Development: An Ethnographic Study*. Empirical Software Engineering. [SJR Q1]
10. Luca Traini, Daniele Di Pompeo, Michele Tucci, Bin Lin, Simone Scalabrino, Gabriele Bavota, Michele Lanza, Rocco Oliveto, and Vittorio Cortellessa. 2022. *How Software Refactoring Impacts Execution Time*. ACM Transactions on Software Engineering and Methodology. [SJR Q1]

INVITED TALK

- 63rd CREST Open Workshop. 2023. University College London, London (UK). website

PROFESSIONAL SERVICE

Journal Peer-review

Regularly serve as a reviewer for several international journals, such as IEEE TSE, ACM TOSEM, EMSE and JSS (see more in ORCID).

MODELLO PER LA PRESENTAZIONE DEL PROGETTO DI RICERCA

Program Committee Member of International Conferences

- IEEE/ACM International Conference on Software Engineering - Research Track. 2026. [Core A*]
- IEEE International Conference on Software Analysis, Evolution and Reengineering - Industrial Track. 2026. [Core A]
- International Conference on Software Maintenance and Evolution – RENE Track. 2026. [Core A]
- International Conference on Evaluation and Assessment in Software Engineering (EASE) - Emerging Results Track. 2024. [Core A]
- IEEE International Conference on Software Analysis, Evolution, and Reengineering (SANER) - Tools Demo Track. 2024. [Core A]
- International Conference on Performance Engineering (ICPE) - Research track. 2023. [Core B]
- International Conference on Performance Engineering (ICPE) - Data Challenge track. 2023. [Core B]

Organizing Committee Member of International Conferences and Workshops

- Co-Organizer of 2nd Workshop on DevOps for Sustainability (DevOpsSustain 2026)
- Co-Organizer of 1st Workshop on DevOps for Sustainability (DevOpsSustain 2025)
- Sponsors Co-Chair of 18th IEEE International Conference on Software Testing, Verification and Validation (ICST 2025)
- Workshops Co-Chair of 16th International Conference on Performance Engineering (ICPE 2025)
- Co-Organizer of 9th Workshop on Challenges in Performance Methods for Software Development (WOSP-C 2024)
- Data Challenge Track co-chair of 15th International Conference on Performance Engineering (ICPE 2024)
- Virtualization Co-Chair, 14th European Conference on Software Architecture (ECSA 2020)

SCIENTIFIC RESPONSIBILITY FOR FUNDED RESEARCH PROJECTS

- [2021 - 2023] Responsible for the activity “Advanced data visualization techniques” (WP3: Research) within the “Territori Aperti” project.

- [2023] Principal Investigator for the Progetto di Avvio alla Ricerca "Enhancing Software Performance Testing using Artificial Intelligence" funded by the University of L'Aquila. *The project led to the publication of the article "AI-driven Java Performance Testing: Balancing Result Quality with Testing Time" in the prestigious ASE conference [Core rank A*].*
- [2024-now] Co-leader of Group-0 "Cross-cutting products for different phenomena" within Spoke 5 of the "ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing"

AWARDS

Best Industrial Paper Award at IEEE International Conference on Software Analysis, Evolution and Reengineering 2024 "RADig-X: a Tool for Regressions Analysis of User Digital Experience".

ASN INDICATORS 2024/26

No. Journal Articles (5 years): 6 - *Citations (Scopus):*112 - *H-Index (Scopus):* 6

e. Eventuali componenti del gruppo di ricerca (solo per i progetti di ricerca di base)

Federico Di Menna, PhD Candidate (Borsa di Ricerca) [INF/01], DISIM
Muhammad Imran, Research Fellow (Borsa di Ricerca) [INF/01], DISIM

f. Settore di ricerca ERC di riferimento per la proposta (indicare anche due sotto-settori)

PE6_3 - Software engineering, operating systems, computer languages

g. Abstract (max 1000 caratteri)

Large Language Models (LLMs) have recently gained prominence for automating software engineering tasks, such as code completion and refactoring (e.g., GitHub Copilot). However, operating these models requires substantial computational resources, which in turn leads to significant energy consumption and carbon emissions. This growing environmental impact conflicts with sustainable computing goals, raising concerns about the long-term feasibility of such AI-driven software development. To address this challenge, AI researchers have proposed various compression strategies to reduce the computational cost of language models. However,

the impact of these strategies in the specific software engineering context remains largely unexplored. Therefore, this research project seeks to systematically evaluate and compare existing compression approaches within software engineering contexts—aiming to reduce computational cost and carbon footprint—and ultimately establish guidelines for achieving greener, more sustainable AI-driven software development.

h. Descrizione del progetto (max 8.000 caratteri, compresi eventuali riferimenti bibliografici. È consentito inserire figure nella proposta. Le figure non concorrono alla determinazione del calcolo del numero dei caratteri.)

a. **Stato dell'arte**

Since its introduction, the transformer architecture [1] has become the de facto standard in language modeling. Transformer-based language models have achieved state-of-the-art performance across numerous natural language processing (NLP) tasks and have recently gained traction in the Software Engineering (SE) field. Over the past few years, large language models (LLMs) have been applied to automate a wide range of SE tasks [2], such as vulnerability detection, code generation, and code completion. For instance, Google claims that these models are now responsible for generating about a quarter of the new code for their products[1].

However, a major obstacle to the practical adoption of LLMs has always been their significant computational cost [3]. To address this issue and increase their sustainability, the AI community has developed various strategies to reduce the computational demands of these models. These strategies have recently begun to gain attention in the SE field. For instance, Shi et al. [4] applied knowledge distillation to drastically reduce the memory size of code models, Wei et al. [5] investigated the impact of quantization on code generation tasks, and D'Aloiso et al. [6] explored the impact of three well-known compression strategies on both the effectiveness and efficiency of a language model across different SE tasks.

These studies have demonstrated that, when carefully selected, these strategies can significantly reduce the computational cost of automating SE tasks. Nevertheless, despite these advancements, there remains a substantial gap in the literature regarding the sustainability of LLMs for SE. Current studies only marginally address energy consumption, often relying on rough estimates rather than rigorous measurements [5,7]. Furthermore, these studies typically focus on older-generation language models [4,5,6,7] (e.g., CodeBERT [8] and CodeT5 [9]), which have been surpassed by more powerful models such as those from OpenAI, Anthropic, Meta, and other leading organizations. This research gap can be explained by the following challenges. First, using these models often requires advanced hardware, which may not be readily available to research laboratories. Second, assessing the energy consumption of LLMs requires careful measurements on isolated execution hardware, a resource rarely available in academic institutions that rely on shared clusters used by multiple research groups.

Consequently, there is still a lack of knowledge on how recent compression strategies can

[1] https://blog.google/inside-google/message-ceo/alphabet-earnings-q3-2024/#full-stack-approach

effectively enhance the energy efficiency of LLMs in SE tasks.

b. **Obiettivi**

Our research goal is to examine how existing compression strategies affect the energy consumption of LLMs within a variety of SE tasks. Drawing on recent findings [4,5,6,7], we anticipate that compression techniques will vary considerably in their impact depending on (i) the specific LLM, (ii) the compression method employed, and (iii) the SE task at hand. Therefore, our aim is to develop guidelines that help practitioners and researchers select the most suitable compression strategy based on their context, thus reducing energy consumption without compromising the model's effectiveness for the target SE task.

c. **Metodologia**

We will begin by conducting a thorough review of the AI literature to identify state-of-the-art compression strategies that can be employed to lower the energy consumption of LLMs. Based on this review, we will then implement the most promising techniques on leading open-source LLMs, such as Llama [10], Qwen [11], and DeepSeek [12].
Next, we will evaluate the resulting "compressed" models on a range of SE tasks (e.g., vulnerability detection, code completion, and code generation). These experiments will be carried out in an isolated environment to (i) measure any changes in the models' effectiveness and (ii) quantify the corresponding gains in energy efficiency.
The requested funding will be critical for acquiring hardware that enables rigorous and reliable energy measurements in a controlled setting. This infrastructure will ensure that external factors are minimized, allowing us to accurately compare energy usage across different compression strategies. Furthermore, we will utilize established SE benchmarks (e.g., CodeXGlue[2], HumanEval[3], and SWEBench[4]) to comprehensively assess both the effectiveness of the "compressed" LLMs.
We plan to make the developed pre-trained LLM models publicly available to facilitate their reuse in practice and foster further research in the field.

d. **Piano di lavoro**

The research project is designed to last for a period of 12 months and is structured into six tasks:

i. *Literature review.* This task consists of surveying the existing body of work to identify promising compression strategies aimed at reducing LLMs' energy consumption.
ii. *LLMs, Benchmarks and experimental environment.* This involves (i) identifying open-source LLMs that can operate within our experimental constraints (e.g., hardware limitations), (ii) selecting a subset of SE tasks of interest and their associated benchmarks, and (iii) configuring the hardware/software environment for the experiments. We plan to use the funds we have obtained to purchase the specialized hardware required for these experiments.
iii. *Implementing Compression Strategies*. In this phase, we will implement the selected compression strategies on the chosen LLMs.
iv. *Evaluation.* This task covers designing and executing the experimental pipeline to evaluate both the effectiveness and energy consumption of the compressed models. For effectiveness, we will rely on widely accepted metrics from SE

[2] https://github.com/microsoft/CodeXGLUE
[3] https://github.com/openai/human-eval
[4] https://www.swebench.com

literature. (For energy consumption, we will utilize dependable tools provided by hardware manufacturers (e.g., nvidia-smi[5]) while maintaining an isolated execution environment.

v. *Results Analysis.* This task involves comparing the effectiveness and energy metrics of the compressed LLMs against their original counterparts. By analyzing these comparisons, we aim to derive guidelines on which strategies are most effective, depending on the specific SE task and LLM considered.

vi. *Dissemination.* This task involves writing the paper that reports the results of the project and disseminating the work at international conferences. Our goal is to submit the resulting article to top-tier software engineering venues, such as ASE, ICSE, or FSE.

The following table outlines the timeline for each task:

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
i. Literature Review												
ii. LLMs, Benchmarks and Exp Env..												
iii. Implementing Compression Strategies												
iv. Evaluation												
v. Results Analysis												
iv. Dissemination												

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS 2017).

[2] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, and H. Wang. 2024. Large Language Models for Software Engineering: A Systematic Literature Review. ACM Transactions on Software Engineering and Methodology (ACM Trans. Softw. Eng. Methodol.).

[3] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni. 2020. Green AI. Communications of the ACM.

[4] J. Shi, Z. Yang, B. Xu, C. Fang, and D. Lo. 2023. Compressing pre-trained models of code into 3 MB. In Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering (ASE '22).

[5] X. Wei, S. K. Gonugondla, S. Wang, S. Cao, and G. Sridhara. 2023. Towards greener

[5] https://docs.nvidia.com/deploy/nvidia-smi/index.html

yet powerful code generation via quantization: An empirical study. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2023).

[6] G. D'Aloisio, L. Traini, F. Sarro, and A. Di Marco. 2025. On the Compression of Language Models for Code: An Empirical Study on CodeBERT. In Proceedings of the IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER'25).

[7] J. Shi, Z. Yang, H. J. Kang, B. Xu, J. He, and D. Lo. 2024. Greening Large Language Models of Code. In Proceedings of the 46th International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS '24).

[8] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, and M. Zhou. 2020. CodeBERT: A pre-trained model for programming and natural languages. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1536–1547. Association for Computational Linguistics.

[9] Y. Wang, W. Wang, S. Joty, and S. C. H. Hoi. 2021. CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021), pages 8696–8708. Association for Computational Linguistics.

[10] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, … and R. Ganapathy. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783.

[11] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, … and T. Zhu. 2023. Qwen technical report. arXiv preprint arXiv:2309.16609.

[12] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, … and Y. Piao. 2024. Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437.

i. Elementi di originalità e innovazione della proposta e impatto in termini di rilevanza dell'avanzamento nella ricerca di base per la comunità scientifica di riferimento (max 3000 caratteri)

The successful execution of this study will result in a paradigm shift in AI-driven software engineering. Enhancing knowledge about the potential implications of compression strategies on energy efficiency will lead to a greater awareness of the possibility of improving the sustainability of LLMs in SE. This has implications for both practitioners and researchers. From a practitioner's perspective, engineers can leverage our guidelines to understand the potential trade-offs between the effectiveness of LLMs and their energy consumption, thereby enabling the sustainable use of LLMs in daily practice. On the researcher's side, the proposed approach can foster further studies on topics such as reproducibility and the development of

more effective strategies for reducing the energy demands of LLMs in SE.
In addition, the proposed study can have a broader impact on software engineering research. Many current areas in software engineering involve the use of LLMs as part of their research. Reducing the computational cost of LLMs will enable researchers to exploit more powerful models within the same hardware constraints and/or reduce the carbon footprint of their studies, thereby enhancing the sustainability of SE research.
The PI has experience in this field, having recently conducted a study [1] investigating the impact of compression strategies on the efficiency and effectiveness of older-generation language models (such as CodeBERT) in various SE tasks, in collaboration with the University College London. The PI is currently engaged in a collaboration with the Simula Research Laboratory to assess the impact of quantization methods in the Automated Program Repair. He has also led research on software efficiency evaluation, serving as the first author on five publications in top-tier software engineering venues [2, 3, 4, 5, 6].

j. Impatto del progetto in riferimento alle tematiche di genere (facoltativo, max 3000 caratteri)

-

Piano di spesa

Voce di spesa	*Importo (Euro)*
Borse di ricerca (art.2 del Regolamento per il conferimento di borse di ricerca attualmente in vigore)	<u>Proroga di 7 mesi borsa di ricerca Muhammad Imran</u> (periodo di proroga giugno-dicembre 2026) L'apporto del borsista sarà essenziale per il raggiungimento degli obiettivi del Progetto, in particolare alla luce dell'esperienza maturata durante il periodo trascorso presso la *Vrije Universiteit Amsterdam*, uno dei principali centri di riferimento a livello mondiale nell'ambito della sostenibilità nell'ingegneria del software. ***Previsione di spesa*: 9.100 €**
Rinnovo assegni di ricerca	-
Materiali di consumo	-
Attrezzature, strumentazioni, software	-

MODELLO PER LA PRESENTAZIONE DEL PROGETTO DI RICERCA

Missioni	Partecipazione al *NII Shonan Meeting* on "*Green Intelligence: Sustainable Development and Operations of Intelligent Systems*" https://shonan.nii.ac.jp/seminars/251 Il meeting, al quale il PI è stato invitato lo scorso dicembre, riunirà ricercatori internazionali attivi nei settori della sostenibilità e ingegneria del software, e sarà dedicato a tematiche strettamente allineate con gli obiettivi del Progetto. ***Previsione di spesa*: 2.500 €**
Acquisto prodotti ritenuti necessari per la realizzazione del progetto (es. materiale librario, licenze per l'accesso a banche dati, ecc.)	-
Pubblicazioni, organizzazione di convegni e workshop	The *2nd International Workshop on DevOps for Sustainability (DevOpsSustain 2026)* https://devopssustain.github.io/ws2026 Il PI è organizzatore, per il secondo anno consecutivo, del workshop *DevOpsSustain*, co-locato con la *ACM International Conference on the Foundations of Software Engineering (FSE),* una delle più prestigiose conferenze nell'ambito del software engineering, che si terrà nel 2026 a Montreal (Canada). Il workshop affronta tematiche strettamente affini al progetto. Il finanziamento consentirà al PI di partecipare in presenza all'organizzazione del workshop. ***Previsione di spesa*: 3.400€**