

1. Proposed Title

Generative AI for Software Engineering (GAIforSE)

2. Proposers

- Davide Falessi, University of Tor Vergata, Rome, Italy, falessi@ing.uniroma2.it, Viale del Politecnico 1, 00133, Rome.
- Douglas C. Schmidt, Vanderbilt University, d.schmidt@vanderbilt.edu, 1025 16th Avenue South, Nashville TN, 37212
- Xin Xia, Huawei Technologies, Hangzhou, China, xin.xia@acm.org, 410 Jianghong Road, Binjiang District, Hangzhou, Zhejiang Province, 310007, China
- Hongyu Zhang, Chongqing University, hongyujohn@gmail.com, 55 Daxuecheng South Road, Chongqing 401331, China

3. Introduction and Relevance

Generative AI (GAI) for software engineering refers to the application of generative models and algorithms in software development, testing, maintenance and evolution. Generative models are machine learning models that can create new data based on a series of inputs such as text, image or code. GAI has the potential to revolutionize the world in which we live, including the way we develop software. GAI can reasonably support software engineers in any task such as requirement gathering or design. For example, GAI can help in the collection of requirements by analyzing inputs from stakeholders and generating initial requirements for the software system. GAI can help in the design phase by generating design artefacts such as wireframes, user interface designs and architectural diagrams from requirements and standards. GAI is a revolutionary technology, so accurately predicting the software engineering tasks it will be useful for is challenging. GAI for Software Engineering is an important topic for IEEE Software readers because it represents a significant change in the way software is developed and maintained. The target audience for this special topic includes software engineers, researchers, and professionals interested in the intersection of AI and software engineering. The special issue will cover a wide range of topics, including but not limited to:

- Generative models and algorithms for code generation, testing, and design
- Integration of GAI in software development workflows
- Evaluation and validation of GAI models for software engineering
- Applications of GAI in software maintenance and evolution
- Ethical and social implications of GAI for software engineering

4. Focus

Although machine learning (ML) has already provided numerous improvements in many areas of software engineering, such as requirements quality, traceability recovery, design assessment and testing priority, GAI differs from ML in many ways. ML algorithms are usually used to classify or predict results based on training data, while GAI algorithms are used to generate new data with characteristics similar to training data. GAI techniques are generally more complex and sophisticated than traditional ML techniques since they require modeling the underlying distribution of the data and generating new data that are coherent and meaningful.

In summary, traditional ML has already provided numerous improvements in many areas of software engineering, but GAI offers a unique approach to generating new data and content that can further improve software development and maintenance processes. Using GAI algorithms, software engineers can automate many tedious and time-consuming tasks and accelerate the pace of software innovation. The focus of the special issue on GAIforSE will be to address the following questions and perspectives:

- What are the current state-of-the-art GAI technologies and tools for software engineering and what are their limitations and challenges?
- What are the best practices for integrating GAI into software development workflows, and what are the potential benefits and risks?
- What are the ethical and social implications of the use of GAI in software engineering and how can we ensure fairness, accountability and transparency?
- What are the emerging trends and opportunities for using GAI in software engineering, such as natural language processing, computer vision and machine learning?
- How can we evaluate and validate GAI models for software engineering, and what should be the metrics and benchmarks we use?

This theme issue does not include software engineering for GAI models since it is a different topic which might require its own venue.

5. Organization

The theme of GAIforSE is:

Introduction: The introduction provides an overview of the theme, defines key concepts and terminology, and explains the relevance of GAIforSE. We will also discuss the challenges and opportunities presented by GAI and highlight some of the main research directions in the field. The introduction will not simply summarize the articles included but will provide a coherent framework for understanding them. In addition, we will include a sidebar with references to provide readers with additional resources.

Sought articles: We plan to include a balanced selection of submitted articles that cover different aspects of GAIforSE that are relevant to practitioners. We will strive to ensure

that the articles reflect the diversity of perspectives and experiences in the field, including academic, industrial and government contributions. We will also consider the inclusion of a point counter article to highlight different views on the subject.

Expected submissions and acceptance: Based on our initial outreach and discussions with potential authors, we expect to receive roughly 15 submissions. We will carefully review and evaluate each submission on the basis of its relevance, originality, technical reliability and clarity of presentation. We anticipate accepting approximately six papers for publication in the theme issue.

6. List of Potential Reviewers and Authors

The following list of academic experts will be contacted when the theme is accepted. All these people are in our personal contact network and we expect our invitation to be accepted with great confidence. In addition, we are planning to invite industrial reviewers, paper by paper, from our extensive industry collaboration network, to review the relevance of this paper.

- Aitor Arrieta, University of Mondragon, Spain
- Alex Serebrenik, Eindhoven University of Technology, Netherlands
- Andreas Zeller, Saarland University, Germany
- Annibale Panichella, Delft University of Technology, Netherlands
- Audris Mockus, University of Tennessee, United States
- Ayse Bener, Ryerson University, Canada
- Bram Adams, Queen's University, Canada
- Christian Bird, Microsoft Research, United States
- Daniela Damian, University of Victoria, Canada
- David Lo, Singapore Management University, Singapore
- Dongmei Zhang, Microsoft, China
- Eleni Stroulia, University of Alberta, Canada
- Emerson Murphy-Hill, Google, USA
- Fabio Palomba, University of Salerno, Italy
- Gabriele Bavota, Università della Svizzera Italiana, Switzerland
- Georgios Gousios, Delft University of Technology, Netherlands
- Harald Gall, University of Zurich, Switzerland
- Jane Cleland-Huang, University of Notre Dame, United States
- Jiayuan Zhou, Huawei, Canada
- Jules White, Vanderbilt University, USA
- Jürgen Cito, TU Wien, Austria
- Kui Liu, Huawei, China
- Li Li, Tsinghua University, China
- Massimiliano Di Penta, University of Sannio, Italy
- Mayank Agarwal, IBM Research, USA
- Mei Nagappan, University of Waterloo, Canada

- Michele Lanza, Università della Svizzera Italiana, Switzerland
- Michael Muller, IBM Research, USA
- Michael Pradel, University of Stuttgart, Germany
- Nachiappan Nagappan, Meta Platforms Inc., USA
- Rafael Prikladnicki, UC Irvine, USA
- Sebastian Proksch, University of Delft, Netherlands
- Steve Versteeg, Microsoft, Australia
- Tao Xie, Peking University, China
- Tom Zimmermann, Microsoft, USA
- Vijayaraghavan Murali, Meta Platforms Inc., USA

7. Promotion Plan

We intend to invite participants from industry and academia using various strategies. We will include an explicit note in the application letter to encourage the submission of documents from industry and academia. We will also contact our industrial partners to submit papers and participate in special publications. We will distribute the special using various channels, including mailing lists(such as SEWorld), social media(such as Twitter, LinkedIn), university contacts, etc. We will also publish special papers at software engineering and AI conferences(such as FSE, ICSE, MSR, ML, ICML). We have excellent contacts with the organizers of these conferences. We will seek their help to promote the event to a wider audience of potential sponsors.

8. Call for Papers

Title: *Generative AI for Software Engineering (GAIforSE)*

Generative AI (GAI) for software engineering refers to the application of generative models and algorithms in software development, testing, maintenance and evolution. Generative models are machine learning models that can create new data based on a series of inputs such as text, image or code. GAI has the potential to revolutionize the world in which we live, including the way we develop software. GAI can reasonably support software engineers in any task such as requirement gathering or design. For example, GAI can help in the collection of requirements by analyzing inputs from stakeholders and generating initial requirements for the software system. GAI can help in the design phase by generating design artefacts such as wireframes, user interface designs and architectural diagrams from requirements and standards. GAI is a revolutionary technology, so accurately predicting the software engineering tasks it will be useful for is challenging. GAI for Software Engineering is an important topic for IEEE Software readers because it represents a significant change in the way software is developed and maintained. The target audience for this special topic includes software engineers, researchers and professionals interested in the intersection of AI and software engineering.

We invite article submissions covering all aspects of GAI for Software Engineering including, but not limited to:

- Cloud computing
- Component-based service-oriented systems
- Configuration management
- Human aspects
- Maintenance and evolution
- Privacy
- Program synthesis & transformations, automated defect repair
- Reliability
- Requirements engineering
- Software architecture and design
- Software engineering processes
- Software product line engineering
- Software security and trust
- Testing, verification, and validation

How-to Submit An Article

– For general author guidelines:

www.computer.org/publications/author-resources/peer-review/magazines

– For submission details: software@computer.org

– To submit an article: <https://mc.manuscriptcentral.com/sw-cs>

Questions?

For more information about the focus, contact the guest editors:

- Davide, Falessi, falessi@ing.uniroma2.it
- Schmidt, Douglas, d.schmidt@vanderbilt.edu
- Xin, Xia, xin.xia@acm.org
- Hongyu Zhang, hongyujohn@gmail.com

Submission Guidelines

Manuscripts must not exceed 4,200 words, including figures and tables, which count for 250 words each. Submissions in excess of these limits may be rejected without refereeing. The articles we deem within the theme and scope will be peer reviewed and are subject to editing for magazine style, clarity, organization, and space. Be sure to include the name of the theme you're submitting for.

Articles should have a practical orientation and be written in a style accessible to practitioners. Overly complex, purely research-oriented or theoretical treatments aren't appropriate. Articles should be novel. *IEEE Software* doesn't republish material

published previously in other venues, including other periodicals and formal conference or workshop proceedings, whether the previous publication was in print or electronic form.

9. Guest Editor (GE) Qualifications

/*****/

Corresponding Editor - Davide Falessi

-Summary: Davide Falessi is an Associate Professor of Software Engineering at the University of Rome Tor Vergata, Italy where he leads the Software Engineering in Rome research group. His main research interest is in devising and empirically assessing scalable solutions for the development of software-intensive systems. He received his Ph.D., MSc, and BSc degrees in Computer Engineering from the University of Rome Tor Vergata, Italy.

-Qualifications as Editor: Dr. Falessi is the Associate Editor in Software Economics of IEEE Software and an Associate Editor of the Empirical Software Engineering Journal, Springer. He is a reviewer board member of the IEEE Transactions on Software Engineering and of the ACM Transactions on Software Engineering and Methodologies. He has been the Guest Editor of two very successful special issues at IEEE Software: "Actionable Analytics for Software Engineering" and "Agile Development at Scale". Dr. Falessi has been the Guest Editor in several journals, including the Empirical Software Engineering Journal and the Journal of Systems and Software.

-Five most relevant publications:

- 1. D. Falessi, S. Laureani, J. Çarka, M. Esposito, D. Dacosta: "Enhancing the defectiveness prediction of methods and classes via JIT", Empirical Software Engineering; An International Journal, 23, 1, 2023.*
- 2. J. Çarka , M. Esposito, D. Falessi, "On effort-aware metrics for defect prediction", Empirical Software Engineering; An International Journal, 27, 6, 2022*
- 3. D. Falessi, A. Ahluwalia, M. Di Penta, "The Impact of Dormant Defects on Defect Prediction: a Study of 19 Apache Projects", ACM Transactions on Software Engineering and Methodology, 31,1, 2022*
- 4. B. Vandehei, D. A Da Costa, and D. Falessi. "Leveraging the Defects Life Cycle to Label Affected Versions and Defective Classes", ACM Transactions on Software Engineering and Methodology 30, 2, 2021*
- 5. D. Falessi, J. Roll, J. LC Guo, and J. Cleland-Huang, "Leveraging Historical Associations between Requirements and Source Code to Identify Impacted Classes." IEEE Transactions on Software Engineering 46(4): 420-441, 2020.*

/*****/

Douglas C. Schmidt

-Summary: Dr. Douglas C. Schmidt is the Cornelius Vanderbilt Professor of Computer Science, Associate Chair of Computer Science, and a Senior Researcher at the Institute for Software Integrated Systems at Vanderbilt University. He is also a Visiting Scientist at the Software Engineering Institute (SEI) at Carnegie Mellon University (CMU), where he served as the Chief Technology Officer from 2010 to 2012.

-Qualifications as Editor: Dr. Schmidt has served as the editor or co-editor of a dozen special issues for various publications, including Proceedings of the IEEE, IEEE Computer, IEEE Network, IEEE Communications, Communications of the ACM, USENIX Computer Systems Journal, Springer Annals of Telecommunication Journal, the International Journal of Network Protocols and Algorithms (NPA), and Wiley's Journal of Theory and Practice of Object Systems.

-Five most relevant publications:

- 1. Quchen Fu, Ramesh Chukka, Keith Achorn, Thomas Atta-fosu, Deepak R. Canchi, Zhongwei Teng, Jules White, and Douglas C. Schmidt. Deep Learning Models on CPUs: A Methodology for Efficient Training, Journal of Machine Learning Theory, Applications and Practice, Vol. 1, pp 45--82, 2023.*
- 2. Quchen Fu, Zhongwei Teng, Marco Georgaklis, Jules White, and Douglas C. Schmidt, NL2CMD: An Updated Workflow for Natural Language to Bash Commands Translation, The Journal of Machine Learning Theory, Applications and Practice, Vol. 1, pp 1--38., 2022.*
- 3. Peng Zhang, Christopher Fannesbeck, Douglas C. Schmidt, Jules White, Samantha Kleinberg, Shelagh A. Mulvaney, Understanding Barriers to Self-Management in Type 1 Diabetes Using Machine Learning and Momentary Assessment, the JMIR Journal of mHealth and uHealth, Vol 10., No 5., March 2022.*
- 4. Douglas C. Schmidt, Jesse Spencer-Smith, Quchen Fu, and Jules White, Cataloging Prompt Patterns to Enhance the Discipline of Prompt Engineering, International Conference on Reliable Software Technologies (AEiC 2023) 13-16 June 2023, Lisbon, Portugal.*
- 5. Ashraf Elnashar, William Schreiber, Jules White, and Douglas C. Schmidt, Question Formulation and Transformer Model Resilience, Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI'22), December 14-16, 2022; Las Vegas, USA.*

/*****/

Xin Xia

Summary: Xin Xia is the Director of the Software Engineering Application Technology Lab at Huawei, China. His research interests include intelligent software engineering, mining software repositories, and empirical software engineering. He has published

more than 280 peer-reviewed conference and journal papers, and he won the ACM SIGSOFT Early Career Researcher Award in 2022.

-Qualifications as Editor: Dr. Xia has served as the editor of many software engineering journals, including ACM Transactions on Software Engineering and Methodology, Empirical Software Engineering (EMSE), Automated Software Engineering (ASEJ), and Journal of Software: Evolution and Process. He has served as the editor or co-editor of many special issues for various publications, including EMSE, ASEJ, Journal of Software and Systems, Information and Software Technology, and Science China: Information Science.

-Five most relevant publications:

1. Yanming Yang, Xin Xia, David Lo, John C. Grundy: A Survey on Deep Learning for Software Engineering. ACM Comput. Surv. 54(10s): 206:1-206:73 (2022)
2. Chao Liu, Cuiyun Gao, Xin Xia, David Lo, John C. Grundy, Xiaohu Yang: On the Reproducibility and Replicability of Deep Learning in Software Engineering. ACM Trans. Softw. Eng. Methodol. 31(1): 15:1-15:46 (2022)
3. Zhongxin Liu, Xin Xia, Meng Yan, Shanping Li: Automating Just-In-Time Comment Updating. ASE 2020: 585-597 (ACM SIGSOFT Distinguished Paper Award)
4. Xing Hu, Ge Li, Xin Xia, David Lo, Zhi Jin: Deep code comment generation. ICPC 2018: 200-210 (ACM SIGSOFT Distinguished Paper Award, GS Citation: 511)
5. Zhongxin Liu, Xin Xia, Ahmed E. Hassan, David Lo, Zhenchang Xing, Xinyu Wang: Neural-machine-translation-based commit message generation: how far are we? ASE 2018: 373-384 (ACM SIGSOFT Distinguished Paper Award)

/*****/

Hongyu Zhang

- Summary: Hongyu Zhang is a Professor and Dean of School of Big Data and Software Engineering, Chongqing University, China. He also worked at the University of Newcastle, Microsoft Research Asia, and Tsinghua University. He received his PhD degree from National University of Singapore in 2003. His research is in the area of Software Engineering, in particular, software analytics, maintenance, fault diagnosis, and intelligent software engineering. The main theme of his research is to improve software quality and developer productivity by mining software data. He has published more than 200 research papers in reputable international journals and conferences. He received five ACM Distinguished Paper awards. He is a Distinguished Member of ACM, a Senior Member of IEEE, and a Fellow of Engineers Australia (FIEAust).

- Qualifications as Editor: Prof. Zhang actively engages with the research community and has been involved in organizing many academic conferences. For example, he was a General co-chair of the 36th International Conference on Software Maintenance and Evolution (ICSME 2020), Program co-chair of the 25th Asia-Pacific Software Engineering Conference (APSEC 2018), Program co-chair of the 18th IEEE International Conference on Software Quality, Reliability, and Security (QRS 2018), and a co-organizer of Dagstuhl Seminar on "Testing and Verification of Compilers" in Dec 2017. He is on the Program Committee of more than 100 academic conferences,

including top conferences such as ICSE, ASE, FSE, ISSTA, ICSME, AAAI, ICML, ICLR, and NeurIPS. He is also an Associate Editor of ACM Computing Surveys, Journal of Automated Software Engineering, and Journal of Systems and Software.

- Five most relevant publications:

- 1. Xiaodong Gu; Hongyu Zhang; Sunghun Kim; Deep Code Search, 40th ACM/IEEE International Conference on Software Engineering (ICSE 2018), Gothenburg, Sweden, May 2018.*
- 2. Jian Zhang, Xu Wang, Hongyu Zhang, Hailong Sun, Kaixuan Wang and Xudong Liu, A Novel Neural Source Code Representation based on Abstract Syntax Tree, In Proc. the 41th International Conference on Software Engineering (ICSE 2019), Montreal, Canada, May 2019.*
- 3. Van-Hoang Le; Hongyu Zhang; Log-based anomaly detection with deep learning - how far are we?, 44th IEEE/ACM International Conference on Software Engineering (ICSE 2022), Pittsburgh, PA, USA, May 2022.*
- 4. Huong Ha; Hongyu Zhang; DeepPerf: Performance Prediction for Configurable Software with Deep Sparse Neural Network, 41st IEEE/ACM International Conference on Software Engineering (ICSE 2019), Montreal, Canada, May 2019.*
- 5. Shuzheng Gao, Hongyu Zhang, Cuiyun Gao, Chaozheng Wang, Keeping Pace with Ever-Increasing Data: Towards Continual Learning of Code Intelligence Models, Proc. 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE), May 2023.*