Mohammad Jalili Torkamani

Computer Science Ph.D. Candidate | AI4SE Researcher | AI Software Engineer

contact@MJalili.com – MJalili.com

Google Scholar – linkedin.com/in/MJaliliT – github.com/MJaliliT

EDUCATION

PhD, Computer Science Jan 2024 – Exp. Dec 2028
University of Nebraska–Lincoln (UNL) GPA: 3.9/4.0
Dissertation: Automated generation of program assertions and verification conditions using generative AI and search-based techniques

Master of Science, Computer Science Jan 2024 – Dec 2025
University of Nebraska–Lincoln (UNL) GPA: 4.0/4.0
Thesis: LLM-assisted CWE identification, severity assessment, and vulnerability description generation

Bachelor of Science, Computer Engineering Sep 2016 – Sep 2020
K. N. Toosi University of Technology (KNTU) GPA: 3.2/4.0

SKILLS

Programming: Python, Java, C/C++, Bash

Software Engineering: Git, Docker, CI/CD, System Design, Program Analysis, Static Code Analysis, Dafny, OOP, Algorithm Design, Data Structures, Scalable System Design, Program Comprehension

AI & Large Language Models: PyTorch, LangChain, Fine-Tuning, Prompt Engineering (Few-shot, CoT/ToT), RAG, Model Training, Model Evaluation, Structured LLM Outputs, Inference Pipelines, Model Deployment

Backend: FastAPI, REST APIs, Microservices, JWT, Nginx

Distributed Systems: AWS, Kafka, Redis, Celery, Asynchronous Programming

Software Security: API Security, Vulnerability Analysis

Software Testing & Reliability: Unit Testing, Fuzzing, Test-Driven Development (TDD), Test Automation, Debugging, Profiling

Databases & Storage: PostgreSQL, MySQL, MongoDB, Schema Design, Data Modeling

OS: Ubuntu Linux, Windows, macOS

ACADEMIC EXPERIENCE

Teaching Assistant **Aug 2026 – Present**
University of Nebraska–Lincoln

- **Courses:** Software Testing, Verification, and Analysis (CSCE/SOFT 467/867); Introduction to Robotics (ROBO 303, ENGR 391-002).
- Support laboratory sessions, classroom activities, and office hours while mentoring students in software testing, program analysis, verification, robotics programming, debugging, and system integration.
- Evaluate assignments, laboratories, projects, quizzes, and examinations and collaborate with instructors on course materials and technical guidance.

Research Assistant **Jun 2026 – Jul 2026**
University of Nebraska–Lincoln

- Engineered scalable Python/Java program-analysis and repository-mining infrastructure to extract, classify, and analyze production assertions across 500+ open-source Java repositories.
- Developed LLM-based pipelines for production-assertion generation and program comprehension using structured prompting, fine-tuning, and automated evaluation.
- Built end-to-end validation infrastructure integrating JavaParser-based static analysis, source-code transformation, compilation, unit-test execution, and statistical evaluation of generated assertions.
- Designed reproducible experimentation workflows and research artifacts supporting empirical AI4SE studies on software quality and developer productivity.

Teaching Assistant **Jan 2026 – May 2026**
University of Nebraska–Lincoln

- **Course:** Introduction to Computer Engineering (CSCE-164)
- Guided laboratory sessions in digital logic, combinational and sequential circuits, Arduino programming, embedded systems, debugging, and hardware–software integration.
- Evaluated laboratory assignments, programming projects, quizzes, and examinations while providing technical support during office hours.

Research Assistant **Jan 2024 – Dec 2025**
University of Nebraska–Lincoln

- Engineered Python- and Java-based frameworks for LLM-assisted vulnerability triage, including CWE identification, CVSS severity assessment, and vulnerability-description generation from source-code changes.
- Designed scalable pipelines for CVE, CWE, and CVSS data preparation and source-code analysis, supporting empirical studies across thousands of vulnerability records.
- Fine-tuned and benchmarked open-source and API-based LLMs across prompting strategies, source-code granularities, and task configurations for vulnerability understanding.
- Automated reproducible evaluation workflows for hierarchical CWE prediction, severity classification, generated-description assessment, statistical analysis, and model comparison.
- Collaborated with interdisciplinary research teams on AI-driven software security, vulnerability analysis, and program-analysis research.

Teaching Assistant **Sep 2017 – Jan 2018**
K. N. Toosi University of Technology

- **Course:** Advanced Programming
- Supported Java laboratory instruction and office hours covering object-oriented programming, debugging, object-oriented design, and software development practices.
- Evaluated programming assignments, projects, and examinations and provided technical feedback.

Teaching Assistant **Jan 2017 – Jun 2017**
K. N. Toosi University of Technology

- **Course:** Fundamentals of Computer Programming
- Supported C/C++ instruction and laboratories covering control flow, functions, arrays, pointers, data structures, debugging, and problem solving.
- Evaluated programming assignments, laboratory exercises, and examinations and provided technical feedback during office hours.

INDUSTRY EXPERIENCE

Software Engineer **Apr 2022 – Jul 2022**
Arzyaban Salamat Pars Company

- Led the design and development of a production-grade internal web application supporting organizational workflows and internal operations.
- Implemented backend business logic, relational database models, and RESTful services, contributing across application architecture, data modeling, and integration.
- Collaborated with cross-functional teams on testing, deployment, and release activities to improve software reliability and maintainability.

Software Engineer **Oct 2018 – May 2020**
Medreek (Ayandeh Sazan) Publication

- Developed an Android application for personalized study planning for high-school students using Java and the Android SDK.
- Implemented recommendation logic and user-facing mobile features to generate individualized study plans and support student decision-making.
- Improved application usability and interaction flows to strengthen the student learning experience.

Founding Software Engineer **Oct 2016 – Oct 2019**
SeedS/Lexeen Software Startup

- Contributed to the early technical direction of the startup while supporting hands-on software development and project-based learning initiatives.
- Led neural-network workshops for 10+ undergraduate students, covering machine-learning fundamentals and practical model development.
- Designed and delivered Java/Android development workshops for approximately 20 undergraduate students and mentored participants in software design, debugging, and mobile application development.

Software Engineering Internship **Jul 2019 – Sep 2019**
Sarmad Insurance

- Developed an Android application for calculating insurance premiums based on business rules provided by Sarmad Insurance.
- Implemented premium calculation logic and intuitive user interfaces using Java and Android SDK.
- Worked closely with company engineers to validate business rules and application functionality.

SELECTED PROJECTS

CAT GAN – WGAN-GP Cat Image Generation catgan.MJalili.com

- Designed, trained, and evaluated a WGAN-GP in Python/PyTorch to generate 128×128 RGB cat-face images from a 200-dimensional latent space using custom convolutional Generator and Critic architectures.
- Implemented GPU-accelerated training with the Wasserstein objective, gradient penalty, multiple Critic updates per Generator update, and Adam optimization.
- Built and deployed a responsive Streamlit application with CPU inference on a VPS; enabled interactive latent-space exploration with persistent state to vary individual dimensions while holding others fixed.

AUTONOMOUS SOFTWARE FACTORY – Multi-Agent Application Creation and Evolution

- Built a platform that turns natural-language product ideas into Docker-deployed applications using Python/FastAPI, Pydantic, SQLAlchemy, Alembic, PostgreSQL/SQLite, and a JavaScript/HTML/CSS dashboard.
- Automated private GitHub synchronization, DNS configuration, Nginx routing, and SSH orchestration; added scheduling, pause/resume, access management, and recovery controls with generated source and execution hosted on the VPS.
- Engineered a hierarchical Codex-based workflow: five parallel specialists assess UX, product value, reliability, design, and security; a judge selects improvements; and a coding agent implements them in a constrained Docker workspace.
- Added structured agent outputs, risk-sensitive deliberation, and weighted quality/latency/cost/risk evaluation; integrated automated tests, security and licensing checks, isolated canary validation, rollback, and evolution-history monitoring.

ASSERTION STUDY – Empirical Study of Production Assertions in Open-Source Java Software

- Designed a large-scale empirical study of production assertions in open-source Java systems, examining their prevalence, structure, evolution, and usage patterns across hundreds of repositories.
- Engineered scalable Python and Java repository-mining and program-analysis pipelines to identify, extract, normalize, classify, and statistically analyze hundreds of thousands of assertion instances.
- Combined repository-scale quantitative analysis with a developer survey to study why developers introduce or avoid production assertions, their intended purposes, and perceived reliability and maintainability tradeoffs.

CASEY – LLM-Assisted CWE Identification, Severity Assessment, and Vulnerability Description Generation

- Engineered an end-to-end Python framework for LLM-assisted software vulnerability triage, jointly targeting CWE identification, CVSS severity assessment, and vulnerability-description generation from source-code changes.
- Built scalable experimentation pipelines for 4,900+ training/validation samples and 800+ unseen vulnerabilities, supporting analysis at file, method, and change-hunk granularities.
- Fine-tuned and benchmarked API-based and open-source LLMs across prompting, fine-tuning, input-granularity, and task configurations, with reproducible evaluation for hierarchical CWE prediction, severity classification, and generated descriptions.

ASSERTIFY – LLM-Based Production Code Assertion Generator

- Engineered an automated LLM-driven framework that analyzes Java methods and generates candidate production assertions with source-level placement information using structured prompting.
- Built a multi-stage validation pipeline integrating structured LLM outputs, JavaParser-based program analysis, source-code transformation, compilation, and automated test execution against real Java projects.
- Designed reusable Python/Java experimentation infrastructure for benchmarking multiple models, prompting strategies, and source-code contexts across assertion validity, placement, and runtime behavior.

KAJAL – AI-Assisted Source Code Grammar Extraction and Program Understanding

- Developed an LLM-based framework for automatically inferring source-code grammar rules from software repositories, investigating whether language models can recover syntactic structure from code examples.
- Engineered modular Python pipelines for source-code sampling, prompt construction, grammar generation, structured-output processing, and evaluation of inferred language representations.
- Investigated LLM-assisted grammar inference as a foundation for downstream program analysis, source-code understanding, documentation, and developer tooling.

SELECTED PUBLICATIONS

- **M.J. Torkamani**, A. Sharma, N. Mehrotra, R. Purandare, "Automatic Generation of Java Production Assertions Using Large Language Models.", *ACM Transactions on Software Engineering and Methodology*, (*Under Review*)
- **M.J. Torkamani**, M. Rashedul Hasan, H. Bagheri, B. Sharif, R. Purandare, "The Impact of Code Granularity on LLM-Based Vulnerability Understanding: CWE Identification, Severity Assessment, and Description Generation.", *ACM Transactions on Software Engineering and Methodology*, (*Under Review*)
- M.S. Bilesavar, N. Mahmoudi, **M.J. Torkamani**, K. Kiashemshaki, "Boosting accuracy and interpretability in multilingual hate speech detection through layer freezing and explainable AI", *International Conference on Natural Language Processing and Machine Learning*, 2026, Published, AR-LPML-SAPL-260826-367.
- **M.J. Torkamani**, J. NG, N. Mehrotra, M. Chandramohan, P. Krishnan, R. Purandare, "Streamlining security vulnerability triage with large language models.", arXiv preprint, 2501.18908.
- K. Kiashemshaki, **M.J. Torkamani**, N. Mahmoudi, "Secure coding for web applications: Frameworks, challenges, and the role of LLMs.", arXiv preprint, 2507.22223.
- **M.J. Torkamani**, N. Mahmoudi, K. Kiashemshaki, "LLM-Driven adaptive 6G-ready wireless body area networks: Survey and framework.", arXiv preprint, 2508.08535.
- K. Kiashemshaki, EN. Chukwuani, **M.J. Torkamani**, N Mahmoudi, "Secure and scalable blockchain voting: A comparative framework and the role of large language models.", arXiv preprint, 2508.05865.
- **M.J. Torkamani**, A. Sharma, N. Mehrotra, R. Purandare, "ASSERTIFY: Utilizing large language models to generate assertions for production code.", arXiv preprint, 2411.16927.
- K. Kiashemshaki, **M.J. Torkamani**, N. Mahmoudi, M.S. Bilehsavar, "Simulating a bias mitigation scenario in large language models.", arXiv preprint, 2509.14438.
- **M.J. Torkamani**, "Kajal: Extracting grammar of a source code using large language models.", arXiv preprint, 2412.08842.

SERVICE & OUTREACH

Volunteer Coding Mentor — SoC CLICKS | *University of Nebraska–Lincoln, 2026*

- Mentored middle-school students in Python programming, computational thinking, and problem-solving activities.
- Supported university-led outreach initiatives promoting computer science and STEM education among K–12 students.

STEM Outreach Volunteer — Bright Lights | *University of Nebraska–Lincoln, 2026*

- Introduced 22 students in a K–8 STEM program to computer science, computational thinking, robotics, and automation concepts through interactive demonstrations.
- Promoted early STEM engagement by demonstrating real-world applications of software systems, automation, and intelligent technologies.
- Facilitated hands-on robotics activities using programmable rover robots for maze-navigation challenges and robotic arms for task-oriented problem solving.

Coding for Kids Outreach Volunteer | *University of Nebraska–Lincoln, 2025*

- Guided children under age 10 through interactive, game-based coding activities introducing basic programming concepts.
- Assisted participants with problem-solving tasks and beginner-friendly computational thinking exercises.

HONORS & AWARDS

- **Ranked in the top 1%** among 162,000+ participants for undergraduate entrance.
- **Awarded full graduate funding** during graduate studies.
- **Awarded full merit-based scholarship** during undergraduate studies.

CERTIFICATIONS

- CITI Program: Social/Behavioral Research Investigators and Key Personnel (Valid: 2026–2029).
- National Science Foundation (NSF) I-Corps Entrepreneurship Program, 2026.
- Neural Networks and Deep Learning, Coursera, 2023.
- AI for Everyone, Coursera, 2022.
- Python for AI, Development, and Data Science, Coursera, 2022.