05.25

Computer

Architectures, Benchmarks, Frameworks, and Predictions

IEEE

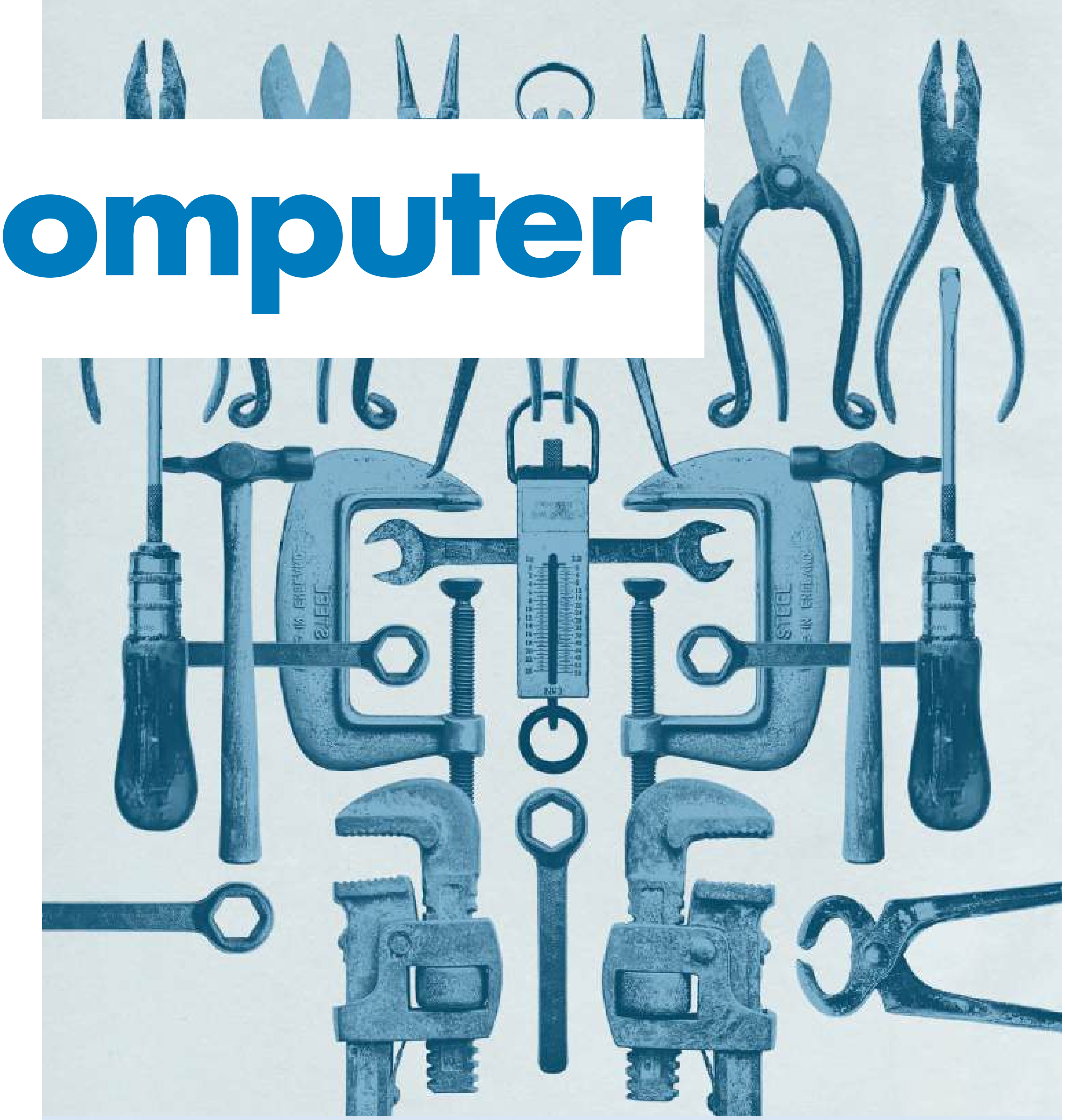
IEEE COMPUTER SOCIETY

vol. 58 no. 5

www.computer.org/computer

Digital Object Identifier 10.1109/MC.2025.3553340

Computer



Digital Object Identifier 10.1109/MC.2025.3534540

MAY 2025

CONTENTS

ABOUT THIS ISSUE ARCHITECTURES, BENCHMARKS, FRAMEWORKS, AND PREDICTIONS

Articles discuss unique technology advancements.

FEATURES CONTINUED

Department

COLUMNS

Membership News

Circulation: *Computer* (ISSN 0018-9162) is published monthly by the IEEE Computer Society. IEEE Headquarters, Three Park Avenue, 17th Floor, New York, NY 10016-5997; IEEE Computer Society Publications Office, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720; voice +1 714 821 8380; fax +1 714 821 4010; IEEE Computer Society Headquarters, 2001 L Street NW, Suite 700, Washington, DC 20036. IEEE Computer Society membership includes a subscription to *Computer* magazine.

Postmaster: Send undelivered copies and address changes to *Computer*, IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Canadian GST #125634188. Canada Post Corporation (Canadian distribution) publications mail agreement number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8 Canada. Printed in USA.

Computer

IEEE COMPUTER SOCIETY http://computer.org // +1 714 821 8380
COMPUTER http://computer.org/computer // computer@computer.org

Digital Object Identifier 10.1109/MC.2025.3534542

ELSEWHERE IN THE CS

Computer Highlights Society Magazines

The IEEE Computer Society's lineup of 11 peer-reviewed technical magazines covers cutting-edge topics ranging from software design and computer graphics to Internet computing and security, from scientific applications and machine intelligence to visualization and microchip design. Here are highlights from recent issues.

computing in SCIENCE & ENGINEERING

Probabilistic Block Term Decomposition for the Modeling of Higher Order Arrays

Tensors are ubiquitous in science and engineering, and tensor factorization approaches have become important tools. The authors of this October–December 2024 *Computing in Science & Engineering* article explore the use of Bayesian modeling in the context of tensor factorization, present a probabilistic extension of the so-called block term decomposition model, and show how it can interpolate between two common decomposition models—canonical polyadic decomposition and Tucker decomposition.

IEEE Annals of the History of Computing

The History of Unigraphics, 1974–2001

Unigraphics was a first-generation commercial computer-aided design and manufacturing system, initially released by United Computing Corporation in 1974. This article, featured in the October–December 2024 issue of *IEEE Annals of the History of Computing*, provides a firsthand account of the evolution of Unigraphics from 1974 through 2001, during which authors Paul Sicking, George Allen, and Wil Valenzuela were members of the development group. The authors describe the motivations behind each major step in the evolution of Unigraphics as well as the underlying technical strategies.

IEEE Computer Graphics AND APPLICATIONS

PerSiVal: On-Body AR Visualization of Biomechanical Arm Simulations

In this November/December 2024 *IEEE Computer Graphics and Applications* article, the authors explore different combinations of techniques for an interactive, on-body visualization in augmented reality of an upper arm muscle simulation model. They focus on a continuum-mechanical simulation model involving five different muscles of the human upper arm, with physiologically realistic geometry. In terms of use cases, the authors focus on the immersive illustration, education, and dissemination of such simulation models. They describe the process of developing six on-body visualization prototypes over a period of five years.

IEEE Intelligent Systems

Knowledge Routing in Decentralized Learning

With mobile and Internet of Things devices becoming pervasive in our lives and recent advances in edge computational intelligence [for example, edge artificial intelligence/machine learning (AI/ML)], it became evident that the traditional methods for training AI/ML models are becoming obsolete, especially with the growing concerns over privacy and security. The authors of this January/February 2025 *IEEE Intelligent Systems* article highlight the key challenges that prevent edge AI/ML from seeing widerange adoption in different sectors, especially for large-scale scenarios.

Digital Object Identifier 10.1109/MC.2025.3549068
Date of current version: 28 April 2025

IEEE Internet Computing

Power-Aware CPU Cap Mechanism in Serverless Computing Environments

Designing elastic resource allocation algorithms for serverless environments is challenging but promising, as slight performance improvements yield significant monetary savings for service providers. Serverless computing supports execution of pipeline operations without explicit resource provisioning. However, effective consolidation strategies are needed to minimize contention among concurrently running applications and to handle workload surges. This article from the November/December 2024 issue of *Internet Computing* presents CPU cap strategies for optimizing power consumption by adjusting CPU voltage and frequency, while adhering to user-defined latency event-driven applications in serverless platforms.

IEEE micro

Monza: An Energy-Minimal, General-Purpose Dataflow System-on-Chip for the Internet of Things

This article featured in the November/December 2024 issue of *IEEE Micro* describes the design and implementation of Monza, a testchip system-on-chip featuring Efficient Computer Company's Fabric processor. The Fabric is a general-purpose, spatial dataflow architecture and compiler designed to minimize energy.

IEEE MultiMedia

Specific Diverse Text-to-Image Synthesis via Exemplar Guidance

This October–December 2024 *IEEE MultiMedia* article investigates an open research task of text-to-image synthesis for generating specific diverse images guided by exemplars. Various conditional generative adversarial networks have been developed to generate images conditioned on the text and add noise for random diversity.

IEEE pervasive COMPUTING

EtherealBreathing: A Holographic Biofeedback Game to Support Relaxation in Autistic Children

The authors of this article in the October–December 2024 issue of *IEEE Pervasive Computing* evaluate a novel biofeedback holographic game, *EtherealBreathing*, designed to support autistic children. In *EtherealBreathing*, children practice box breathing to collect virtual elements to maintain the Earth's balance, using a wearable sensor to measure chest expansion for breath detection.

IEEE SECURITY & PRIVACY

The Path to Autonomous Cyberdefense

Defenders are overwhelmed by attacks against their networks, which will only be exacerbated as attackers leverage artificial intelligence to automate workflows. In this article, featured in the January/February 2025 issue of *IEEE Security & Privacy*, the authors propose a path to autonomous cyberagents able to augment defenders by automating critical steps in the cyberdefense lifecycle.

IEEE Software

A Deep-Learning-Based Visualization Tool for Air Pollution Forecasting

The authors of this article from the March/April 2025 issue of *IEEE Software* present a comprehensive visualization tool that integrates real-time observation and sensing data with various forecasting models, including numerical and deep-learning approaches. The developed software framework efficiently manages data flow, configures forecasting models, and visualizes monitoring and prediction information.

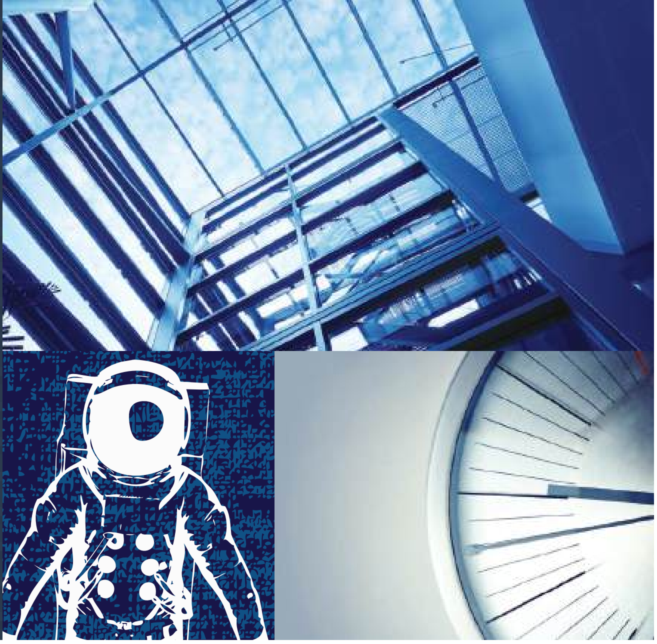
IT Professional

Landscape and Taxonomy of Prompt Engineering Patterns in Software Engineering

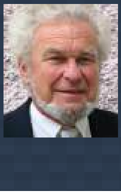
Advancements in large language models (LLMs) have enhanced their ability to handle ambiguous user instructions. However, effective prompt patterns remain crucial for usability and comprehension. This January/February 2025 *IT Professional* article presents a taxonomy of prompt engineering patterns for software engineering. It is based on a systematic literature review that was conducted in early 2023, when LLMs still faced significant limitations in context length and inference capabilities.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *Computer* does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society. All submissions are subject to editing for style, clarity, and space.

Digital Object Identifier 10.1109/MC.2025.3553341



50 & 25 YEARS AGO



EDITOR ERICH NEUHOLD
University of Vienna
erich.neuhold@univie.ac.at

MAY 1975

https://www.computer.org/csdl/magazine/co/1975/05

Guest Editor: Software Engineering; Raymond T. Yeh (p. 15): "To explain what software engineering is about, we need to examine its goals, as well as the fundamental issues and principles derivable from those goals. This is indeed what this special issue of Computer attempts to do." *[Editor's note: In 1975 "software engineering," as a term, was still new enough to warrant such an explanation. The following three articles attempt to do so, and I will extract from them.]*

Software Engineering: Process, Principles, and Goals; Douglas T. Ross et al. (p.17): "We will discuss these issues in terms of four fundamental goals: modifiability, efficiency, reliability, and understandability as well as seven principles that affect the process of attaining these goals: modularity … abstractions … hiding … localization … uniformity … completeness … confirmability."(p. 18) "Despite the obvious differences among these activities, we believe each reflects a common pattern which we call the fundamental process." (p. 19) "This process consists of five basic steps: purpose … concept … mechanism … notation … usage. … The interpretation of the fundamental process is clearly highly context dependent." *[Editor's note: These three types of principles are then arranged as an 140 (4 × 5 × 7) elements block where each element represents one of the issues of software development. Via an example, a few of these issues are then explained. In my honest opinion, these details are too complicated and, in hindsight, they have never been used in such detail.]*

The Software Factory; Charles A. Harlow et al. (p. 28): "That the typical software development process today is without sound engineering basis is perhaps nowhere better illustrated than in a recent study published in IEEE Transactions on Computers." (p. 30) "The Software Factory consists of an integrated and extensible facility of software development tools that supports a recommended methodology. The Factory is designed to operate on a host machine and use the facilities of the host operating system." *[Editor's note: A lengthy but still interesting article that describes the issues of software development both from the technical as well the management side, and tries to provide tools (integrated management, project analysis, and control technique, i.e., IMPACT), to support all of those.]*

A Survey of Analytic Models of Rollback and Recovery Strategies; K. M. Chandy (p. 40): "Systems of intrinsically unreliable components can be made more reliable by introducing redundancy into the system." (p. 41) "Rollback and recovery is a method of enhancing the reliability of systems. The objective function and constraints for rollback and recovery strategies may vary substantially from system to system. … Thus, static checkpointing is appropriate for data base systems while dynamic checkpointing is more appropriate for process control systems." (p. 42) "There is no single model which is applicable to all systems. Three models (A, B, C) on data base systems are considered next: Young described model A and models B and C are described in Reference 3. Process control models are briefly reviewed later. It must be emphasized that this paper outlines only the modeling approach; the reader is directed to more detailed sources for more information." *[Editor's note: This interesting article describes in detail recovery and rollback strategies for database systems, including checkpoints and transaction consistency considerations.]*

Special Feature: An Automated Chinese Telephone Directory; Y. H. Chin et al. (p. 49): "A real-time, on-line information storage and retrieval system for telephone information service has been designed at the Telecommunication Laboratories of Taiwan." (p. 51) "In this system we have two data bases: a telephone directory data base and a Chinese character pattern data base." (p. 54) "By comparison of the slowest system's response time (0.24 sec) with the quickest manual response time, when some of the most frequently used Chinese characters (about 100) are made core-resident, the system's response time will be improved to 200 times

Digital Object Identifier 10.1109/MC.2025.3544118
Date of current version: 28 April 2025

faster than that of a manual system." *[Editor's note: An interesting article that is mostly concerned with the efficient handling of the Chines character of the Taiwan telephone system.]*

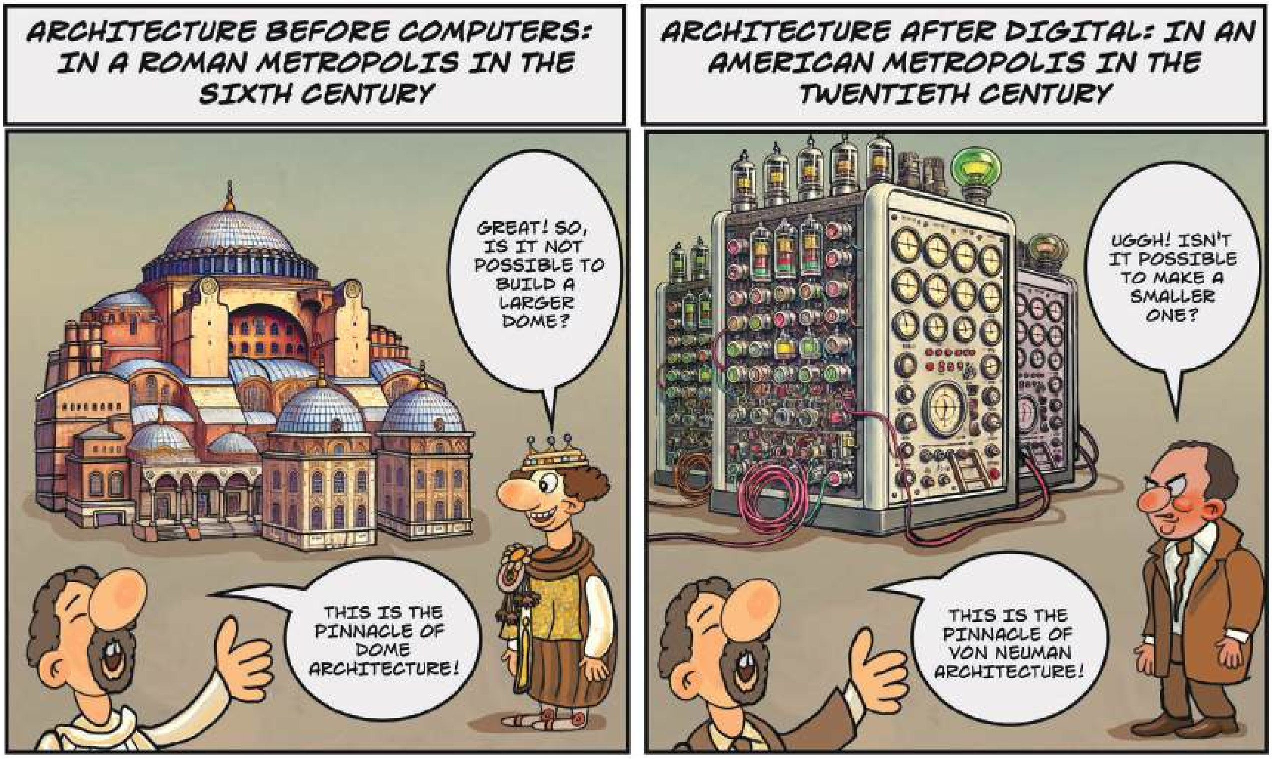
MAY 2000

https://www.computer.org/csdl/magazine/co/2000/05

The Future of the OS for Internet Applications; Reed Hellman (p.12): "One of the key platforms is the operating system that is used for Web servers, e-commerce servers, e-mail servers, and other Internet-based operations. ... While the main focus of Internet related OS activity is on Unix and Windows, a growing number of organizations are using the

COMPUTING THROUGH TIME ARCHITECTURE

BY ERGUN AKLEMAN
ERGUN.AKLEMAN@GMAIL.COM


ARCHITECTURE BEFORE COMPUTERS: IN A ROMAN METROPOLIS IN THE SIXTH CENTURY
GREAT! SO, IS IT NOT POSSIBLE TO BUILD A LARGER DOME?
THIS IS THE PINNACLE OF DOME ARCHITECTURE!
ARCHITECTURE AFTER DIGITAL: IN AN AMERICAN METROPOLIS IN THE TWENTIETH CENTURY
UGGH! ISN'T IT POSSIBLE TO MAKE A SMALLER ONE?
THIS IS THE PINNACLE OF VON NEUMAN ARCHITECTURE!

FOR CENTURIES, NO ONE WAS ABLE TO CONSTRUCT A LARGER DOME THAN HAGIA SOPHIA IN ISTANBUL DUE TO A COMBINATION OF ENGINEERING, MATERIAL, AND STRUCTURAL LIMITATIONS. HAGIA SOPHIA'S DESIGN WAS AN UNPRECEDENTED ACHIEVEMENT, UTILIZING PENDENTIVES TO SUPPORT ITS MASSIVE DOME ON A SQUARE BASE. IT WAS NOT UNTIL THE CONSTRUCTION OF ST. PETER'S BASILICA (1506-1626) AND THE FLORENCE CATHEDRAL (1436) THAT ARCHITECTS, EQUIPPED WITH RECENT ENGINEERING ADVANCEMENTS, WERE ABLE TO BUILD DOMES OF COMPARABLE OR GREATER SIZE.

THE VON NEUMANN ARCHITECTURE, PROPOSED IN 1945, REVOLUTIONIZED COMPUTING WITH THE STORED-PROGRAM CONCEPT, WHERE DATA AND INSTRUCTIONS SHARE MEMORY. THE EDVAC (ELECTRONIC DISCRETE VARIABLE AUTOMATIC COMPUTER), DEVELOPED AT UPENN'S MOORE SCHOOL, WAS THE FIRST COMPUTER DESIGNED UNDER THIS MODEL AND COMPLETED IN 1949. UNLIKE EARLIER ENIAC (ELECTRONIC NUMERICAL INTEGRATOR AND COMPUTER), WHICH USED DECIMAL ARITHMETIC AND MANUAL REPROGRAMMING, EDVAC EMPLOYED BINARY ARITHMETIC, SEQUENTIAL MEMORY, AND STORED INSTRUCTIONS, GREATLY IMPROVING EFFICIENCY.

Digital Object Identifier 10.1109/MC.2025.3548738
Date of current version: 28 April 2025

open-source Linux OS in their servers." *[Editor's note: The interesting short article then compares in detail Windows and Solaris as the main contender and mentions Linux only in passing. As such, it misses the issues that made Linux next to Windows the successive one. Of course, 25 years ago a smart phone operating system (OS) was not of real concern. That only changed in 2008 when Linux based Android started to conquer the world.]*

Will WAP Deliver the Wireless Internet?; Neal Leavitt (p. 16): "Providing this access is particularly challenging for handheld devices … Providing this access is particularly challenging for handheld devices because of their small screens, low memory and power, … Now, however, proponents and many industry observers are touting WAP (the Wireless Application Protocol) as the technology that will become the standardized basis and future of the mobile Internet." (p. 19) "However, WAP will have many users, at least for a couple of years, until more mainstream protocols, like HTML and XML, can be used effectively with mobile devices." *[Editor's note: An article with pros (mostly) and cons for WAP that also contains a correct prediction of the slow demise of WAP around 2009.]*

News Briefs; Ed: Anne C. Lear (p. 21ff): Vendors Begin to Adopt 'Next-Generation' Internet Protocol: Networking hardware and software vendors are slowly beginning to incorporate Internet Protocol version 6 (IPv6) into their products, … However, ISPs have been slow to adopt IPv6 for several reasons. Technically complex and costly, upgrading existing equipment and software will require supporting both IPv4 and IPv6 in hardware and protocol stacks." *[Editor's note: Even now, 25 years later, IPv6 is not universally adopted.]* "Road to Linux Acceptance Could Fork: The issue is becoming important as Linux use in server and other systems increases. Nearly a quarter of the server systems shipped in 1999 included Linux." *[Editor's note: Of course this did not happen. Many applications are using the Linux kernel and thus ensure some interoperability.]*

Gaining Intellectual Control of Software Development; Barry Boehm et al. (p. 27): "Yet despite its critical importance, software remains surprisingly fragile. Prone to unpredictable performance, dangerously open to malicious attacks, and vulnerable to failure at implementation despite the most rigorous development processes, in many cases software has been assigned tasks beyond its maturity and reliability. … After many meetings with the research community, funding agencies, industry, and the public at large, PITAC *[Editor's note: President's Information Technology Advisory Committee]* issued a hard-hitting report that recommended increasing government-sponsored IT research by $1.37 billion annually within five years." (p. 33) "The NSF *[Editor's note: National Science Foundation]* has taken the first steps toward grasping the reins of software development and regaining control not only of the development process but also of the role software will play in our future." *[Editor's note: The article is based on two workshops and lists many concerns about software, software quality, and potential risks. It is quite optimistic but, as we now know, those problems are still around us today.]*

The Push to Make Software Engineering respectable; Gilda Pour et al. (p. 35): "A recognized engineering profession must have an established body of knowledge and skill that its practitioners understand and use consistently. After 30 years, there is still a wide gap between the best and the typical software engineering practices." (p. 37) "The societies, via member involvement, must strive to create a broader consensus on the core of the SE *[Editor's note: software engineering]* profession. Perhaps the SE community needs to address issues more aggressively. … The codification proclaims what is unique about the profession, demarcates the boundaries with related professions, and significantly aids education, certification, and licensing." (p. 41) "Increased collaboration between academia, engineering institutes, and industry would substantially reduce serious mismatches in expectations." *[Editor's note: How interesting that this article and the one above (50 years ago) cover mostly the same issues. Not much seems to have changed in the 25 years between and it is my guess that not much has changed another 25 years later, namely today.]*

What Knowledge Is Important to a Software Professional?; Timothy C. Lethbridge (p. 44): "Efforts to develop licensing requirements, curricula, or training programs for software professionals should consider the experience of the practitioners who actually perform the work. … We used the responses to the 75 questions in our survey to develop three sets of data: the importance of various topics taught in computer science, software engineering, and computer engineering curricula, the emphasis educational institutions place on these topics, and what practitioners believe they currently know about the topics." (p. 45) "We received survey responses from 186 participants with a wide variety of backgrounds." (p. 50) "Conversely, the large amount of on-the-job learning—and greater importance relative to amount known—suggest that educational institutions should place considerably more emphasis on teaching topics such as people skills, software processes, human-computer interaction, real-time system design, and management." *[Editor's note: A very detailed and interesting analysis that could have been used to influence institutional teaching. In my mind, some of it may now be included in what is now termed "computational thinking."]*

Keeping Up with the Changing Web; Brian E. Brewington et.al. (p. 52): "Most information depreciates over time, so keeping Web pages current presents new design challenges. "(p. 53) "Search engines strive to keep track of the ever-changing Web by finding, indexing, and reindexing pages. How should we invest observation resources to keep users happy? … This involved processing nearly 200 gigabytes of HTML data (about 100,000 Web pages per day). The archived information includes • the last modified time stamp, • the time of observation, and • stylistic information (content length, number of images, tables, links, and similar data)." *[Editor's note: The interesting article then continues with a detailed analysis of when webpages have to be reindexed, using the above information as an indication that the content of a webpage has changed.]*

Advances in Network Simulation; Lee Breslau et al. (p. 59): "The Virtual Inter-Network Testbed (VINT) project has enhanced its network simulator and related software to provide several practical innovations that broaden the conditions under which researchers can evaluate network protocols. … The Virtual Inter-Network Testbed (VINT) project provides improved simulation tools for network researchers to use in the design and deployment of new wide-area Internet protocols." *[Editor's note: The article explains how various simulators, but mostly ns, that is an existing simulator, can be used successfully inside of VINT.]*

Practical Verification of Embedded Software; Jørgen Staunstrup et al. (p. 68): "The compositional backward technique is a new algorithm that dramatically improves runtimes compared with the algorithms traditionally used for exhaustive verification." (p. 69) "We use a state machine model that describes a computation as transitions between a fixed set of states. The visualState tool is a conceptually simple state machine model that has received widespread practical use." *[Editor's note: The interesting article describes in detail via examples how the visualState tool can be used for imbedded software verification, but also points to some problems with backward iterations and reachability.]*

Value and Productivity in the Internet Economy; Anitesh Barua et al. (p. 102): "While the Internet is often considered to be a sales and marketing channel, we take the position that it has created complete electronic economy that is already large and growing rapidly, creating new opportunities and jobs." (p. 104) "Digital products companies … offer content and services directly over the Internet. … In contrast, e-tailers sell physical products such as toys, jewelry, and electronics on the Internet that are then shipped to consumers." *[Editor's note: The article then concludes that e-tailers will benefit less than digital product sellers. Of course, that has proven wrong as the article mostly ignores the effect the Internet has on the internal processes of such corporations.]*

An Integrated Architecture for Cooperative Sensing Networks; Jon Agre et al. (p. 106): Distributed sensor networks (DSNs)—consisting of many small, low-cost, spatially dispersed, communicating nodes — have recently been proposed. … Several technical challenges must be overcome to fully realize the viability of the DSN concept in realistic application scenarios." (p. 107) "Such a complex system must be easily deployable, able to self-organize into a functioning network, accommodate random node spacing, self-locate, and identify information destinations such as end users." *[Editor's note: This short article identifies many applications of DSNs, both militarily and civilian, but it does not see the innumerable applications in the smart-phone world. However, some of them, for example, intelligent traffic monitoring, are still outstanding.]*

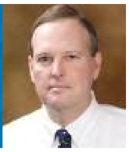
Using Technology and Innovation to Simulate Daily Life; Michael Macedonia (p. 110): "I knew something was up when I saw my daughter shouting at our computer, scolding one of her Sims—a simulated male who kept making mess of his house. … Thus, the primary skill you need to play the game is the ability to plan and queue instructions for your Sims." (p. 111) "Despite all its technical bells and whistles, The Sims' ultimate beauty lies in its ability to immerse you in the Sim world and captivate you with each Sim's autonomy." *[Editor's note: Despite the enthusiasm conveyed in this article, the Sims World never became a public rage but, in my mind unfortunately, it became essential in today's military simulations of strategies, tactics, and when guiding executions.]*

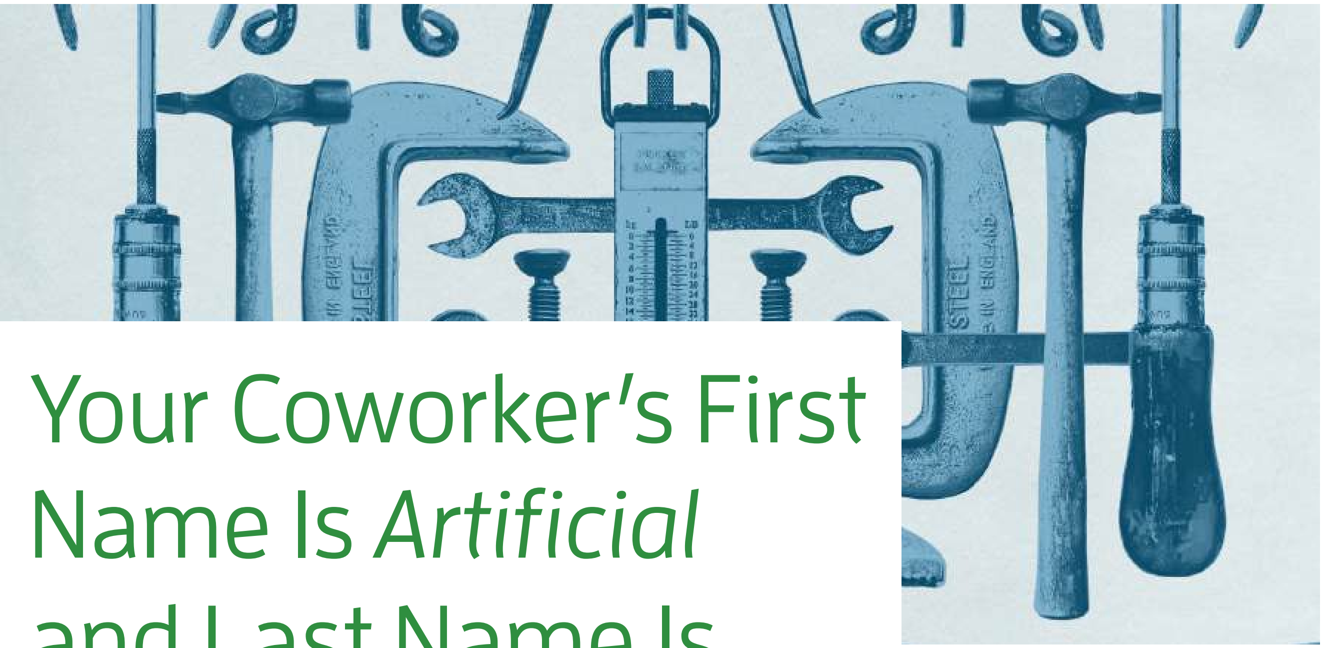
XML: An Interview with Peter Flynn; Ed: Charles Severance (p. 113): "As such, XML is becoming an important tool in application integration. Peter Flynn was a key voice in developing the standard and remains an active observer of how XML is finding its place in the world. … If business can actually agree on what constitutes an invoice order, we might actually see some real XML-enabled e-commerce." *[Editor's note: This interesting article is mostly concerned with the need for universal standardization of application style sheets. Of course, that never happened and still XML and its variants are practically behind every page on the Web.]*

Software Technologies: Fundamental Ideas and Change; Michael Lutz (p. 115): "Welcome to the first installment of the new Software Technologies department. … I'll give you some essential information and I'll present you with technologies that you otherwise might have missed. In Software Technologies, we will explore everything from the mainstream to the marginal." *[Editor's note: I am eager to see and extract for my readers the interesting issues that may be described.]*

Why Funny Money Will Have the Last Laugh; Ted Lewis (p. 112): "Several upstart e-commerce companies are poised to radically alter the abstract entity we call money. … Regardless of what happens to these established banking systems, our definition of money will be radically altered. … **VIRAL CASH** … X.com's PayPal.com service appears to have developed a hypereffective viral marketing model." (p. 110) "CyberWallet required a client-side piece of code, only to find that few people wanted such sensitive code running loose on their disk." *[Editor's note: Here the author has some things right and some wrong. PayPal, created in 1998, is here to stay. People seem to be very willing to put software on their computer/tablet/smartphone (application programming interfaces)! Digital wallets are slow in coming and the pyramid-scheme–based cybercurrencies have lost serious applications but have, in my mind, become just speculation items.]*

EDITOR IN CHIEF JEFFREY VOAS
IEEE Fellow; j.voas@ieee.org





Your Coworker's First Name Is *Artificial* and Last Name Is *Intelligence*

Joanna F. DeFranco, The Pennsylvania State University

Jeffrey Voas, IEEE Fellow

This short message explores how the artificial intelligence revolution is creating both fear and excitement for the workforce.

We've been hearing that people are afraid of losing their livelihoods and ability to work because artificial intelligence (AI) is about to swoop in and take their jobs. Some pundits agree—some do not. We thought we'd take a quick look here at what is myth and what is truth.

To begin, understand that we *are* in another revolution—a paradigm shift—a disruptive change. Whatever you want to call it, it is a shift in how work gets done. The current revolution can be referred to as the *AI Revolution*. It's a shift like the Agricultural, Industrial, and Digital revolutions of the past (Table 1). These "revolutions" changed the way society "worked." They offered new automation that changed the workforce.

Today's AI revolution feels a bit more unsettling. Although this AI revolution evolved from its predecessor "digital revolution," it's more disruptive. One reason is that its scope covers multiple sectors. Another reason is the pace of innovation. AI innovation leads to faster breakthroughs. Generative AI (GenAI) has disrupted the way people work; it will continue to impact the future of work.

HOW IS AI IMPACTING THE CURRENT WORKFORCE?

Some media outlets indicate jobs in "IT" have been trending down with a blame on AI.[1] Along the same

DISCLAIMER

The authors are completely responsible for the content in this message. The opinions expressed here are their own.

Digital Object Identifier 10.1109/MC.2025.3541760
Date of current version: 28 April 2025

IN THIS ISSUE

In the May issue, we include five articles. Each discusses unique technology advancements.

The authors of the first article[A1] integrate federated learning (FL) and blockchain technologies to build an adaptable and secure knowledge-defined networking (KDN) system. The article discusses enhancements to network performance by using self-learning, self-adapting, and self-adjusting capabilities in dynamic and decentralized networks. This new architecture, KDN-FLB, addresses concerns with knowledge sharing and privacy preservation. The article discusses 1) constituents, 2) architectures, 3) processes, and 4) use cases of KDN-FLB.

In the second article,[A2] the authors introduce "DPSmartCity," a context-aware dynamic software-defined networking (SDN) framework that preserves the privacy of Internet of Things (IoT) data for smart cities. The approach uses an SDN controller that employs contextual information and performs trust assessments to determine secure routes from IoT devices to cloud services. In this approach, when a lack of trust is detected, a controller dynamically readjusts the network. The authors argue that this is an improvement over existing approaches (while admitting that there are limitations for certain types of attacks).

In the third article,[A3] the authors introduce Embench IOT 2.0 and DSP 1.0, two new benchmarks for embedded computing. Embench is an improved benchmark over two other existing benchmarks: CoreMark and Dhrystone. The article offers examples of questions that Embench IOT 2.0 and DSP 1.0 benchmarks can answer that these two other benchmarks cannot. The article states that because Embench IOT 2.0 and DSP 1.0 are representative suites of programs that execute quickly, there may be other applications of the technology.

In the fourth article,[A4] the authors focus on advances in computer vision and machine learning that have improved monetary currency (banknotes) recognition. The article discusses deep learning using the Vision Transformer (ViT) architecture. The article evaluates the ViT model on Indian currency denominations as well as four other datasets. The approach is compared with models like ResNet, VGG, GoogleNet, and EfficientNet. The article also discusses the challenges of currency recognition in aiding the visually impaired.

In the final article,[A5] the authors focus on the benefits of fault detection before deployment. The article introduces online failure prediction (OFP), a technique that predicts incoming, immediate failures. OFP allows for preemptive measures to avoid, or at least mitigate, negative consequences. This article shows how recent advances in OFP have made it possible to develop more accurate failure predictors. The authors state that this can allow software developers to create and deploy failure prediction mechanisms throughout the development lifecycle.

I thank the authors for their patience. I hope you enjoy the entire issue.

—Jeffrey Voas, Editor in Chief

APPENDIX: RELATED ARTICLES

A1. Y. Li, P. K. Donta, X. Wang, I. Murturi, M. Huang, and S. Dustdar, "KDN-FLB: Knowledge-defined networking through federated learning and blockchain," *Computer*, vol. 58, no. 5, pp. 16–26, May 2025, doi: 10.1109/MC.2024.3471984.

A2. M. Gheisari et al., "A flexible software-defined networking-based privacy-preserving method for Internet of Things-based smart city environment based on the neighbors situation," *Computer*, vol. 58, no. 5, pp. 27–36, May 2025, doi: 10.1109/MC.2024.3506700.

A3. D. Patterson et al., "Embench IOT 2.0 and DSP 1.0: Modern embedded computing benchmarks," *Computer*, vol. 58, no. 5, pp. 37–47, May 2025, doi: 10.1109/MC.2024.3511352.

A4. D. B. Gajjar, P. Faldu, D. R. Kothadiya, A. P. Chaudhari, and N. M. Bhatt, "DeViTC: Deep-vision transformer to recognize originality of currency," *Computer*, vol. 58, no. 5, pp. 48–56, May 2025, doi: 10.1109/MC.2024.3514151.

A5. J. R. Campos, E. Costa, and M. Vieira, "Predicting failures in complex systems," *Computer*, vol. 58, no. 5, pp. 57–64, May 2025, doi: 10.1109/MC.2025.3526342.

Digital Object Identifier 10.1109/MC.2025.3530201
Date of current version: 28 April 2025

lines, the World Economic Forum's Future Job Report[2] also indicated a workforce reduction due to AI.

> "40% of these employers anticipate reducing their workforces where AI can automate tasks."

> "Clerical and Secretarial Workers—including Cashiers and Ticket Clerks, and Administrative Assistants and Executive Secretaries—are expected to see the largest decline in absolute numbers. Similarly, businesses expect the fastest-declining roles to include Postal Service Clerks, Bank Tellers and Data Entry Clerks."

However, statements like these can be misleading since jobs *are* available—they may simply require a *different skill set*. There are demands for AI and machine learning specialists. The current overarching employment themes are: 1) automation-driven job displacement, 2) a need for workers to reskill, 3) a rise of technology-centric careers, and 4) the transformational impact of automation on businesses.

Is this a surprise? Technology workers have always had to reeducate. As a personal example of graduating in the early 1990's and having to take Fortran as a programming language in my undergraduate electrical engineering degree—the first task I (Joanna) had on an internship was to convert a Fortran program into C. Fortran was outdated before I graduated. Then in my first graduate degree in computer engineering, I was told I needed to know Java to do the homework, so I bought a book (for example, *online educational resources still in the future*). But I switched jobs with that new Java skill I just acquired and named my salary. The point is, if you enter a field of technology, expect to continually add to your skill set—or be out of a job. The difference now is that AI no longer only applies to the tech sector.

AI AFFECTS KNOWLEDGE WORK

AI can automate and digitalize human tasks; however, at this stage, AI is more of a tool to enhance human activities. However, we cannot dismiss a future of *working alongside* AI collaborators, assistants, and tutors.

> The point is, if you enter a field of technology, expect to continually add to your skill set—or be out of a job.

Here's an easy way to think about this. Remember when traveling to an unknown destination might involve using an online mapping service (for example, MapQuest) and printing out directions? Now, we receive advanced directions from crowdsourced GPS applications that know where you are and that can generate real-time directions based on traffic, hazards, road closures, and so on. So, at this stage of the AI revolution, it's like we are in a "MapQuest transition stage."

Besides travel, there are other areas where AI has made an impact. AI can quickly analyze large datasets to identify patterns. This is useful for marketing and medical devices. AI algorithms can aid in personal finance management. AI-based medical devices can manage chronic conditions.

In short, AI impacts almost every sector of knowledge work by automating routine tasks, enhancing decision-making, and enabling professionals to focus on more complex, creative, and strategic aspects of their work. While AI will replace jobs, it opens opportunities for workers to improve productivity and their skills.

WHAT'S NEXT?

Near-term AI advancements emphasize the need for workers and organizations to prioritize *education* and *adaptability* to survive in the AI Revolution.

Reports show that "workers can expect that two-fifths (39%) of their existing skill sets will be transformed or become outdated over the 2025–2030 period." [2] According to the World Economic Forum's job report,[2] robots and autonomous systems are predicted to transform 58% of businesses, while energy generation and storage technologies are expected to impact 41%. However, AI and information processing

TABLE 1. Past revolutions and their changes.

Revolution	Transformation example
Agriculture	Transition to farming, farming tools which led to the growth of civilizations.
Industrial (first and second)	Hand production to machines to large-scale manufacturing (that is, steam engines, coal mining, trains, cars, and so on).
Digital	Automation of information (that is, analog to digital). Widespread personal adoption of computers, mobile phones, the Internet and transforming the way we communicate, market, learn, take care of our health and so on. The emergence of technologies such as Internet of Things, AI, blockchain, and so on.
AI	Automation of intelligence and decision making (that is, virtual assistants, self-driving cars). Revolutionizing industries such as health care, manufacturing, and finance.

technologies are anticipated to have the most significant effect, with 86% of respondents expecting these technologies to transform their businesses by 2030.

AI and information processing technologies are anticipated to have the most significant effect, with 86% of respondents expecting these technologies to transform their businesses by 2030.

And at the time of this writing, we leave you with a few last thoughts:

> "AI, data, and cloud rank highest among in-demand skills, according to Revature data."[3]

> "The push toward AI tools is also reframing how specific categories of IT operate. Software development will experience significant change in the coming years, as Gartner predicts the majority of developers will need to upskill by 2027 due to generative AI."[3]

> "Meta Speeds up AI Hiring While Cutting Thousands of 'Low Performers'"[4]

> "The data shows a shift in cognitive effort as knowledge workers increasingly move from task execution to oversight when using GenAI," the researchers wrote. "Surprisingly, while AI can improve efficiency, it may also reduce critical engagement, particularly in routine or lower-stakes tasks in which users simply rely on AI, raising concerns about long-term reliance and diminished independent problem-solving."[5]

In summary, AI will impact everyone, whether working or nonworking. Workers will have to adjust. Hopefully, AI won't make us dumber, as some suggest.

So, if your new coworker is named Artificial Intelligence, will you be welcoming?

REFERENCES

1. B. Lin, "IT unemployment hits 6% amid overall U.S. jobs growth," *The Wall Street Journal*, Sep. 7, 2024. Accessed: Feb. 10, 2025. [Online]. Available: https://www.wsj.com/articles/it-unemployment-hits-6-amid-overall-u-s-jobs-growth-bc2f2915?mod=article_inline
2. World Economic Forum (WEF), "Future of jobs report," Jan. 2025. Accessed: Feb. 10, 2025. [Online]. Available: https://reports.weforum.org/docs/WEF_Future_of_Jobs_Report_2025.pdf
3. R. Torres, "IT leaders turn to upskilling to close looming skills gap," *HR Dive*, Feb. 6, 2025. Accessed: Feb. 10, 2025. [Online]. Available: https://www.ciodive.com/news/upskilling-technology-data-AI-revature/739367/
4. J. Mann, "Meta speeds up its hiring process for machine-learning engineers as it cuts thousands of 'low performers'," *Business Insider*, Feb. 10, 2025, Accessed: Feb. 10, 2025. [Online]. Available: https://www.businessinsider.com/meta-speeds-up-ai-hiring-while-cutting-thousands-low-performers-2025-2
5. E. Maiberg, "Microsoft study finds AI makes human cognition 'Atrophied and Unprepared'," *404 Media*, Feb 10, 2025. Accessed: Feb. 10, 2025. [Online]. Available: https://www.404media.co/microsoft-study-finds-ai-makes-human-cognition-atrophied-and-unprepared-3/

JOANNA F. DEFRANCO is an associate professor of software engineering at The Pennsylvania State University, University Park, PA 16802 USA, and an associate editor in chief of *Computer*. Contact her at jfd104@psu.edu.

JEFFREY VOAS, Gaithersburg, MD 20899 USA, is the editor in chief of *Computer*. He is a Fellow of IEEE. Contact him at j.voas@ieee.org.

IEEE

IEEE COMPUTER SOCIETY

hosted at

YORK UNIVERSITÉ UNIVERSITY

IEEE COMPSAC

2025

July 8-11
Toronto, Canada

Harnessing the Power of Intelligent Systems: Shaping the Future

https://ieeecompsac.computer.org/2025/

The field of computing is rapidly evolving, driven by groundbreaking advancements in artificial intelligence, machine learning, and data analytics. At the forefront of this transformation lie intelligent systems, poised to revolutionize industries and societies. The 2025 IEEE COMPSAC conference in Toronto will be a global forum to explore intelligent systems' latest breakthroughs and applications across diverse domains.

COMPSAC 2025 will include:

- Keynotes by 2025 Computer Soceity President & CEO Hironori Washizaki, and Yale Patt from UT-Austin, as well as panel discussions on critical emerging topics, such as ethical intelligent systems
- A dynamic symposium program covering deployment of intelligent systems that address real-world challenges. Emerging technologies such as quantum computing and blockchain will also be highlighted, as they hold the potential to unlock new frontiers in computing and engineering by enabling more powerful simulations, secure data sharing, and resilient systems.
- A vibrant workshops program for exploration of development in state-of-the-art research topics.
- A Mentorium session to support the vital development of early-career researchers by fostering mentorship, networking, and professional growth.

2025 Key Organizers

Standing Committee Chair
Sorel Reisman, California State University

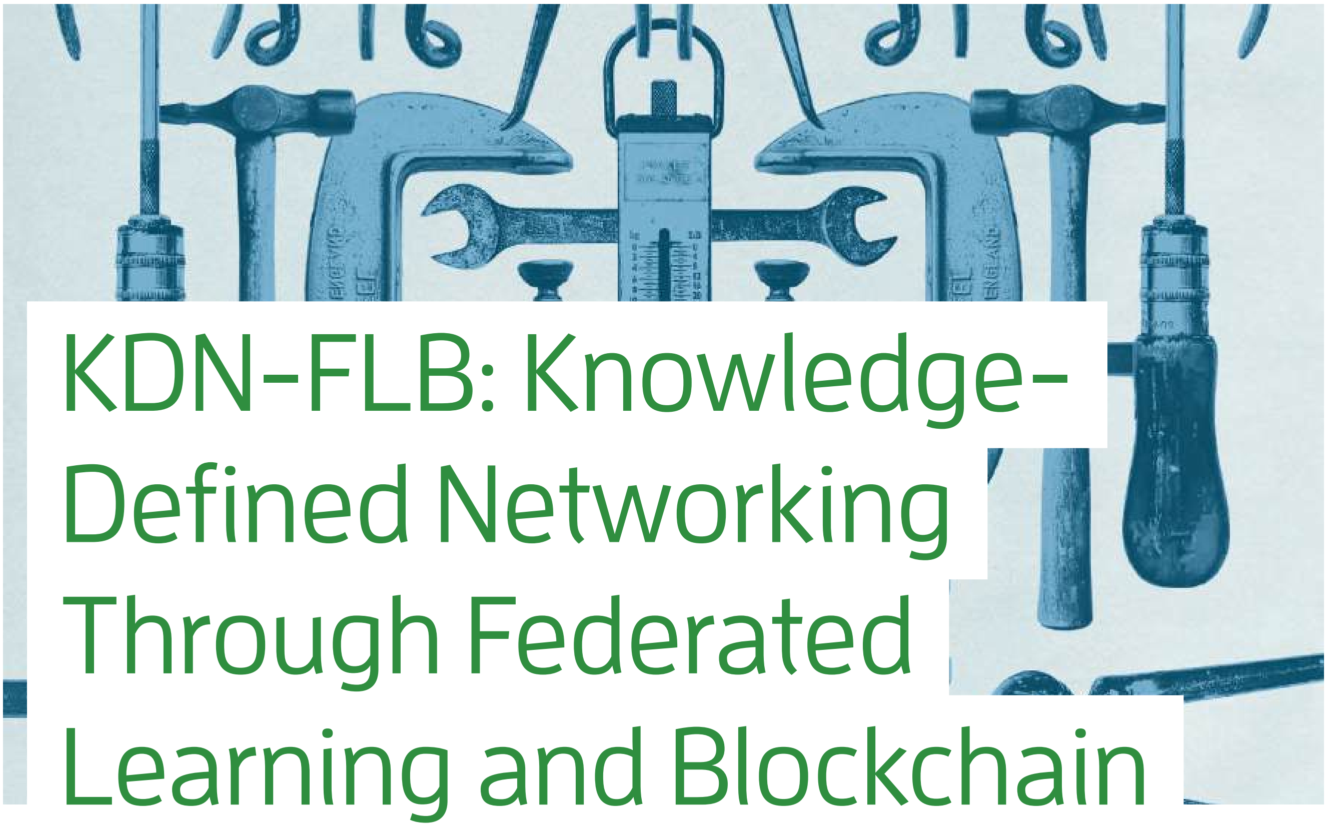
Standing Committee Vice Chairs
Sheikh Iqbal Ahamed, Marquette University
Mohammad Zulkernine, Queen's University
Dave Towey, University of Nottingham Ningbo China (vice chair appointee)

General Chair
Kostas Kontogiannis, York University

Program Chairs in Chief
Hiroyuki Ohsaki, Kwansei Gakuin University
Stelvio Cimato, University of Milan
Miriam Capretz, Western University
Shamem Ahmed, Western Washington University

Workshop Program Chairs
AKM Jahangir Alam Majumder, University of South Carolina Upstate
Munirul Haque, Butler University
Tomoki Yoshihisa, Shiga University
Alfredo Cuzzocrea, Unical

Digital Object Identifier 10.1109/MC.2025.3552995



KDN-FLB: Knowledge-Defined Networking Through Federated Learning and Blockchain

Ying Li, College of Computer Science and Engineering, Northeastern University and TU Wien

Praveen Kumar Donta, TU Wien and Stockholm University

Xingwei Wang, College of Computer Science and Engineering, Northeastern University

Ilir Murturi, TU Wien

Min Huang, College of Information Science and Engineering, Northeastern University

Schahram Dustdar, TU Wien

This article investigates the integration of federated learning and blockchain (FLB) technologies in developing a secure and adaptable knowledge-defined networking system, KDN-LB. It highlights KDN-FLB's potential to enhance network performance and privacy while addressing the challenges of knowledge sharing in decentralized environments.

In this article, we explore the opportunities and benefits of integrating federated learning and blockchain (FLB) technologies to build an adaptable and secure knowledge-defined networking (KDN) system. Our aim is to enhance network performance by ensuring self-learning, self-adapting, and self-adjustment capabilities in dynamic and decentralized network environments. The proposed conceptual architecture, KDN-FLB, also strategically addresses critical challenges in knowledge sharing and privacy preservation within network environments. We discuss the constituents, architecture, processes, and use cases of

Digital Object Identifier 10.1109/MC.2024.3471984
Date of current version: 28 April 2025

KDN-FLB in contemporary networking applications. Additionally, we analyze the benefits, challenges, and future prospects associated with KDN-FLB, making it more intelligent for large-scale, dynamic, and decentralized network environments.

KDN SYSTEMS

The rapid growth of the Internet of Things (IoT) has profoundly expanded the Internet's scale, resulting in increased dynamism and complexity in its applications. For these networks to remain effective, self-learning, self-adaptation, and self-adjustment capabilities are essential, and KDN can fulfill these needs.[1] KDN integrates software-defined networking (SDN) with artificial intelligence (AI), aiming at efficient network management and control (illustrated in Figure 1). KDN incorporates a knowledge plane (KP) into traditional SDN architectures to empower networks to autonomously learn from data, adapt to changing conditions in real time, and optimize performance. On the other hand, machine learning (ML) and AI excel at tracking uncertain and dynamically evolving behaviors, rapidly adapting to changing network conditions, and even resolving issues autonomously. Nevertheless, existing research is mostly fragmented across various aspects of networks, generally addressing specific issues in isolation without comprehensive integration, resulting in two major drawbacks. First, ML lacks interpretability, operating without clear understanding; second, it does not facilitate the aggregation of knowledge for global cognitive reasoning. Also, the current network infrastructure involves both physical and logical distributed resource allocation, creating an urgent need for distributed ML,[2] which FL can effectively address.[3]

In tackling the aforementioned challenges in the literature, limited efforts have been made in advancing KDN. Zhang et al.[4] introduced an advanced Deep-Q network (DQN) routing algorithm enhanced with graph recurrent neural networks (GRNNs) to support intelligent routing decisions within KDN environments. Their approach involved a comprehensive workflow that included developing a network architecture in Mininet, extracting features using GRNNs, and employing DQNs for dynamic path selection. It is necessary to verify the computational efficiency and robustness of this work. Rafiq et al.[5] presented a self-driving system based on KDN that leverages graph NNs (GNNs) to optimize service function chaining deployment and reactive traffic routing across edge clouds, ensuring efficient resource allocation and performance indicator estimation within an SDN framework. Pham et al.[6] explored the application of deep reinforcement learning (DRL) with convolutional NNs within KDN to significantly enhance QoS-aware routing performance, addressing complex network challenges and improving routing configurations in environments with multiple coexisting flows.

He et al.[7] introduced MPDRL, a novel approach that combines DRL with a GNN structure. Based on experiments on the topologies of Internet service provider networks, this approach successfully solves routing optimization problems in dynamic network environments. Another notable contribution comes from Lu et al.,[8] who proposed a blockchain-enhanced FL framework for beyond-5G networks, addressing

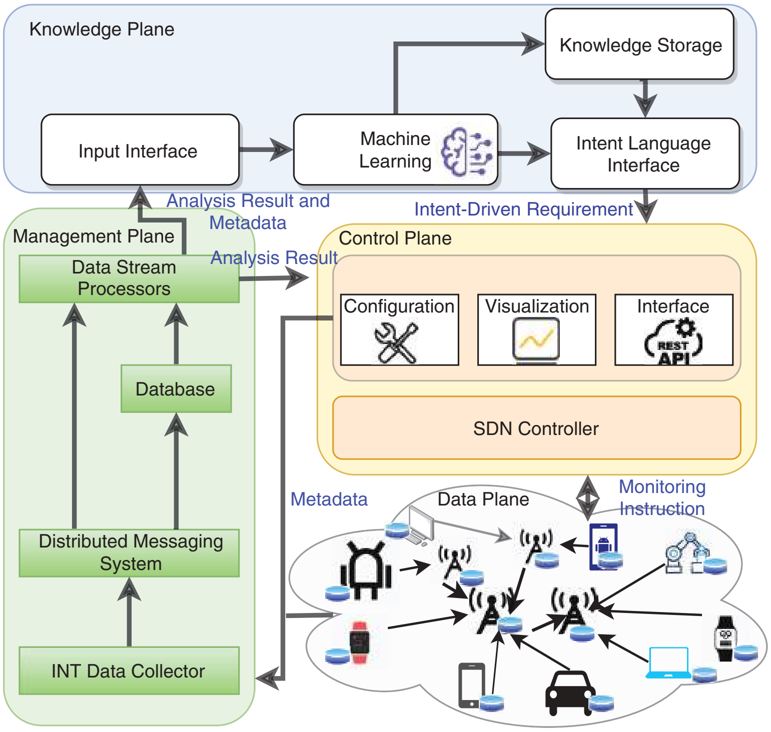


FIGURE 1. Knowledge-defined networking architecture. INT: in-band network telemetry.

security, privacy, and resource optimization through DRL. Despite significant advancements in KDN, FL, and blockchain technologies individually, there is a noticeable lack of comprehensive integration among them in the literature.

As KDN, FL, and blockchain integrate within network systems, they promise security and privacy for knowledge sharing, ownership, and collaboration. Their overarching goal centers on enhancing network systems' performance, imbuing them with self-learning, self-adaptation, and self-adjustment capabilities. In this context, we propose a novel reference framework called *Knowledge-Defined Networking Through Federated Learning and Blockchain* (*KDN-FLB*) to enhance large-scale and dynamic network performance, fortify security measures, and empower the network with self-learning, self-adaptation, and self-adjustment capabilities. The main contributions are summarized as follows:

- We provide a strong motivation for integrating KDN, FL, and blockchain to gain more benefits through KDN-FLB reference architecture.
- We discuss KDN-FLB reference architecture fundamentals, including its architecture, processes, and potential use cases in contemporary networking.
- We employ traffic engineering use cases to evaluate the performance of the proposed KDN-FLB and confirm its superiority.
- We further provide a set of open challenges to implement and extend KDN-FLB for next-generation internet-based applications.

MOTIVATION

A primary goal of KDN is to integrate knowledge across multiple network nodes, facilitating comprehensive global cognitive reasoning and thereby improving overall network performance. This initiative aims to enhance the synergy among distributed nodes, fostering a collective cognitive capability that contributes to an efficient and optimized network.

Integrating FL into KDN is imperative due to the unique challenges in distributed knowledge environments. This multifaceted integration addresses privacy preservation, collaboration augmentation, and distributed knowledge utilization. FL serves as a robust solution to inherent privacy concerns, mitigating breach risks and aligning seamlessly with KDN's distributed nature. In addition to fostering collaboration and sharing knowledge, FL promotes collective intelligence while safeguarding the privacy of individual nodes. Moreover, FL resolves the challenges posed by centralized approaches, making it easier to assemble and utilize distributed knowledge. This integration enables local learning and model updates, optimizing network performance, enhancing adaptability, and ensuring that knowledge remains where it is generated. But there is also a range of key challenges, including covering security and privacy issues in knowledge sharing, knowledge ownership, and collaboration.

Fortunately, blockchain technology presents immense potential due to its decentralization, immutability, openness, transparency, and traceability characteristics, providing innovative solutions to the aforementioned issues. Blockchain's decentralized nature mitigates single points of failure, enhancing system stability and participant control over knowledge resources. Its nontamperability, openness, and transparency establish a robust foundation for knowledge dynamics, ensuring credibility and authenticity. Blockchain's traceability strengthens knowledge credibility and origin scrutiny, fostering trust in knowledge sources within the KDN. It is vital to use these mechanisms to establish trust among KDN contributors and consumers.

KDN-FLB: CONSTITUENTS, ARCHITECTURE, PROCESSES, AND USE CASES

In this section, we discuss the components, architecture, and processes of the proposed KDN-FLB conceptual architecture.

Constituents

KDN-FLB is a multifaceted conceptual architecture that combines several entities to enable decentralized privacy-preserving knowledge sharing. These entities work together to facilitate efficient and secure comprehensive knowledge integration and informed decision making while protecting individual data. Each of the following constituents plays its individual role in the KDN-FLB architecture.

Distributed networks. The KDN-FLB architecture incorporates distributed networks consisting of diverse computing elements such as IoT devices, the edge, or even a computing continuum. These elements typically perform various computational tasks such as data processing, FL model training, FL model aggregation, validation, and blockchain operations.

Participants. The KDN-FLB architecture encompasses several types of participants: individual end users, organizations, and system administrators. End users utilize FL models and knowledge-sharing capabilities to gain insights, make informed decisions, or generate suggestions. Organizations may contribute data, resources, or expertise to the system and interact with FL models and knowledge-sharing processes. System administrators oversee and maintain the technical infrastructure of the KDN-FLB system, ensuring its overall well-being through system updates, security measures, and troubleshooting.

Miners. Miners play a pivotal role in the KDN-FLB architecture by maintaining the blockchain. They validate transactions, create transaction blocks, and secure the network through cryptographic processes such as proof of work or proof of stake.

Proposed KDN-FLB conceptual architecture

The KDN-FLB architecture consists of three components: client-side software, server-side software, and blockchain-side components, as illustrated in Figure 2. On the client side, the user interface facilitates user interactions with the KDN-FLB system, providing visualization tools, controls, and feedback mechanisms for managing FL and blockchain processes. FL data collection involves gathering and preparing local data from individual clients, including user interactions, client-specific information, and other relevant data. Local clients contribute by uploading local model updates without raw data and participating in FL, interacting with server-side components.

The server side, typically in distributed networks, is managed by an FL server that coordinates the FL process, communicates with client-side clients, aggregates local model updates, and securely updates the global model. In hierarchical FL, edge servers can serve as intermediate aggregation servers. Communication middleware on the server side ensures secure data transmission through encryption, authentication, and other security measures. Blockchains receive global knowledge aggregated from distributed clients through the middleware.

On the blockchain side (typically managed by miners), employing a consensus mechanism is crucial for maintaining unanimity on the blockchain state across all nodes, preserving the integrity of the distributed ledger. Historical blockchain data support intelligent decision making, with the decision-making process transmitted to the intent language interface. This interface translates instructions into an imperative language for users to execute, provides feedback to the client side, and enhances system performance.

Processes

This section delineates the working process of the proposed KDN-FLB conceptual architecture, with Figure 3 depicting the detailed process.

Data collection. The initiation phase involves collecting data from dispersed clients, potentially including edge devices, IoT devices, and contributions from participants.

FL model training. Guided by the control plane, FL trains local models by using distributed data, facilitating model training without sharing raw data.

Model aggregation and updates. Under the control plane, diverse clients' trained local model updates are aggregated to generate a global model, which is then disseminated back to each client.

Data security and transparency. Blockchain is used for data security and transparency by recording model training procedures and outcomes, mitigating tampering risks, and providing traceability. However, challenges such as scalability limitations and poor storage extensibility arise due to the blockchain consensus protocol, affecting data safety and reliability. KDN-FLB integrated blockchain involves building a private blockchain and connecting it to a public blockchain to address these issues.

Knowledge extraction. The KP plays a crucial role in deriving meaningful insights from FL models. It involves discerning and extracting valuable knowledge embedded within the aggregated global model. In this phase, KDN-FLB uses FL to extract overall insights from various clients while integrating blockchain to ensure knowledge authenticity and transparency. Therefore, it fosters decentralized intelligence while maintaining data privacy, which is reinforced through blockchain's secure and immutable ledger, which addresses privacy concerns. Also, KDN-FLB's dynamic adaptation utilizes blockchain's immutable recordkeeping to secure historical insights, enabling FL to learn from past experiences and optimize over time. Adapting to evolving network conditions through continuous knowledge extraction improves operational efficiency and user experience.

Intelligent decision making. Intelligent decision making utilizes extracted knowledge to guide strategic choices within distributed networks. Integrating FL and blockchain ensures that decisions are intelligent, privacy preserving, and secure. FL's integration with knowledge extraction supports decentralized decision making, where each client contributes insights from local data, fostering a dynamic distributed decision-making process adaptable to varying conditions. Adaptive decision making is enabled by continuous knowledge extraction, allowing the network to adaptively respond to dynamic environmental changes. The immutable record of historical decisions on the blockchain allows decision makers to refine and optimize future decisions based on past outcomes. Furthermore, blockchain technology ensures the integrity of decision-making processes by providing a decentralized and tamper-resistant ledger for decision records. Bringing FL and blockchain together enhances the security and trustworthiness of decision outputs, establishing a reliable framework for strategic network decisions.

Use cases

In this section, we explore the most appropriate use cases that demonstrate

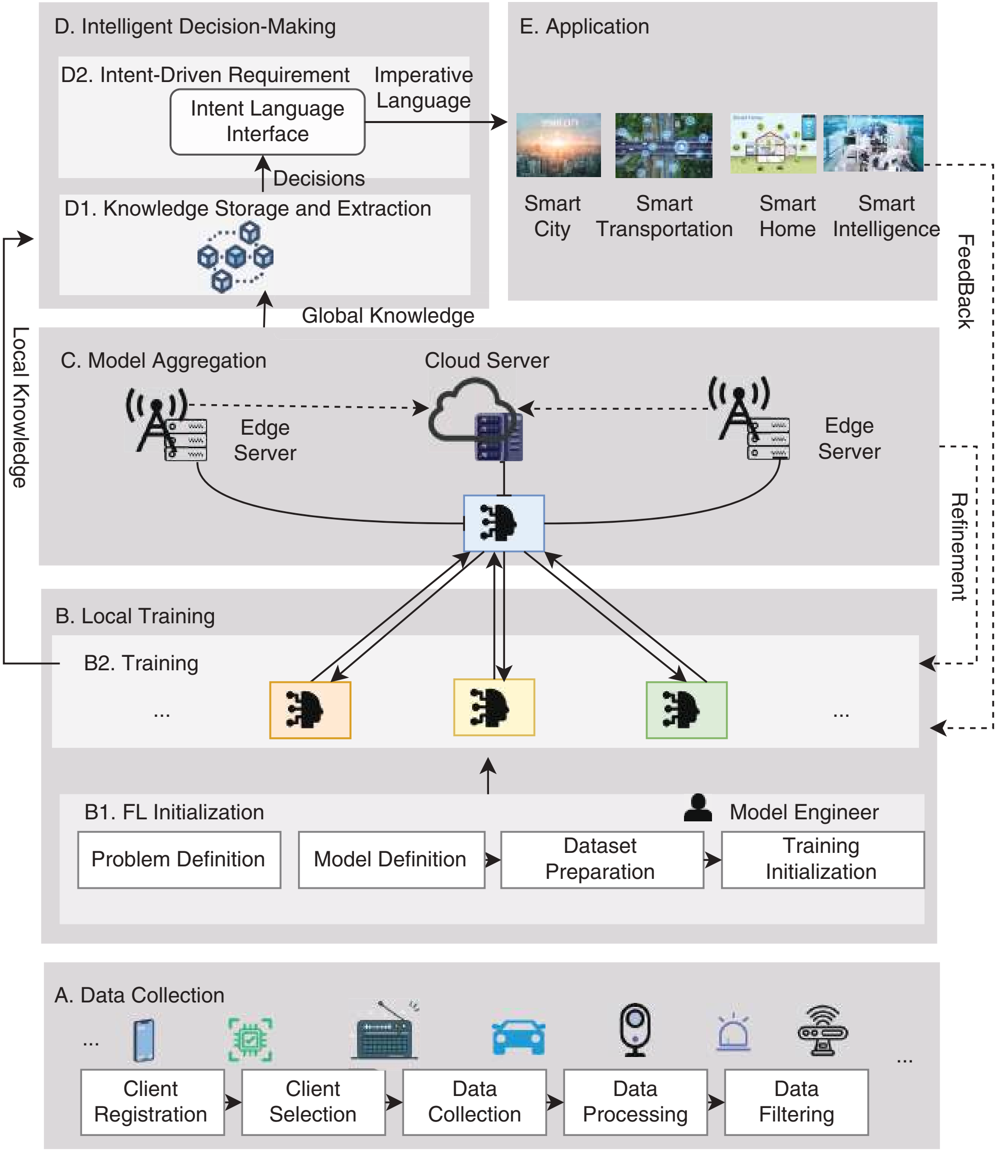


FIGURE 2. The proposed KDN-FLB conceptual architecture.

the effectiveness and usefulness of the KDN-FLB conceptual architecture in addressing real-world challenges.

Traffic engineering. Traffic engineering optimizes telecommunications network performance and efficiency through the strategic control of data, voice, and video traffic. This discipline is essential for effectively utilizing network resources, minimizing congestion, and meeting service quality objectives. Traditional methods often lack intelligence, making it challenging to classify and control incoming traffic based on existing features. Therefore, AI methods such as GNN or multiagent reinforcement learning are considered optimal for early traffic classification, enhancing scheduling and load balancing in dynamic distributed networks to mitigate congestion.[9] In the context of KDN-FLB, historical knowledge trained by FL can be analyzed and stored on the blockchain to learn patterns and relationships between network traffic load and various factors. It facilitates proactive network optimization and enhancements by enabling more accurate traffic load predictions.

Network anomaly detection. Network anomaly detection is critical for identifying and addressing abnormal behaviors in networks. Traditional methods face challenges due to dynamic network changes and the likelihood of false positives or negatives, leading to misinterpretations and ineffective responses. KDN-FLB will be a robust solution for network anomaly detection since it combines the benefits of FL and blockchain. KDN-FLB enhances anomaly detection accuracy by combining historical data from distributed networks with intelligent learning,

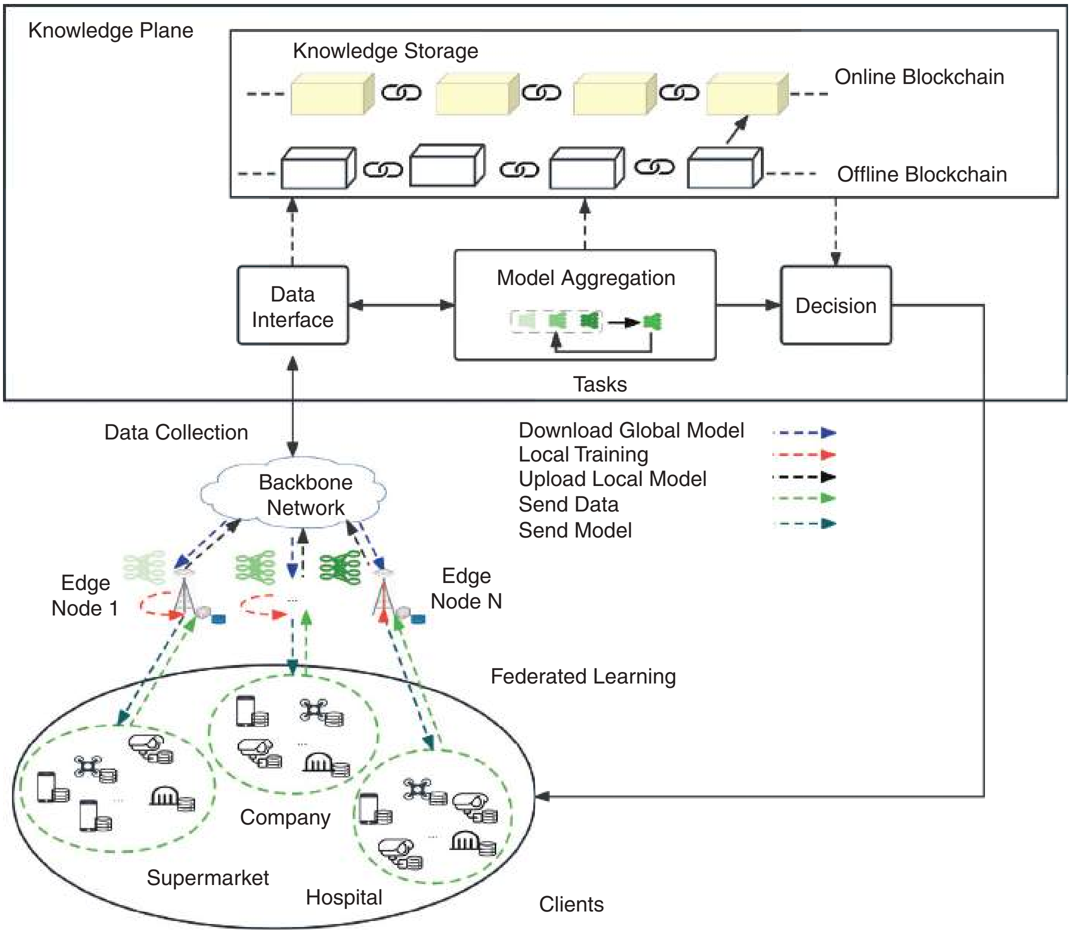


FIGURE 3. The working process of KDN-FLB.

ensuring proactive and secure network management.

Supply chain transparency. Supply chain transparency ensures the clarity and accessibility of information throughout the entire supply chain, from raw material procurement to product delivery, providing stakeholders, including consumers, with precise details about goods' origins, manufacturing processes, and distribution channels. Contemporary supply chains, characterized by intricacies and fragmentation, raise challenges to reliable product tracking and monitoring. Restricted visibility and data silos hinder accurate inventory tracking. KDN-FLB uses blockchain to establish traceability and provenance through the creation of an immutable ledger of supply chain events. KDN-FLB also facilitates compliance and audit efforts by building trust between supply chain clients. In addition, it can enhance security by decentralizing data storage and automating decision-making processes. In summary, KDN-FLB provides a robust architecture for efficient supply chain transparency, ensuring efficiency, security, and trust.

EXPERIMENTS

We employ traffic engineering use cases to evaluate the performance of the proposed KDN-FLB. Existing long- and short-flow classification research relies heavily on static thresholds, frequently resulting in high error rates due to the dynamic nature of network traffic. This article introduces a dynamic coarse-grained classification method based on KDN-FLB to address the complexities and variations in network conditions. The scheduling module subsequently uses the classification results to optimize traffic management, reduce packet loss, and improve transmission stability.

Dataset

Flow size is a key criterion for classifying long and short flows. Al-Fares et al.[10] defined a long flow as a flow that consumes more than 10% of the total link capacity regardless of its duration, which is one of the most important characteristics of flow scheduling. We perform data analysis on the ISCX2016 dataset, revealing that up to 90% of flows have a size smaller than ψ MB. Based on flow size, the classification of flow types is outlined in Table 1.

To expedite coarse-grained long- and short-flow classification, this study utilizes the FlowMeter tool to extract flow information from the first three packets in the dataset, producing a CSV file with 41,816 records and 64 features each. Due to the long-tail distribution, the dataset exhibits sample imbalance, which was addressed using SMOTE, resulting in a balanced dataset of 75,172 records. The data are divided into seven periods, with each period generating 10,738 records. Here, the random forest was selected as the ML algorithm for flow classification.

TABLE 1. The classification of flow types by flow size.

Flow size	Category	Classification
< ψ MB	0	Short
≥ ψ MB	1	Long

Comparison of experimental schemes

Static model long- and short-flow classification. Currently, most schemes for long- and short-flow classification rely on static threshold division. In coarse-grained classification using random forest under static threshold conditions, a model is first trained on existing data to distinguish between long and short flows. This model is then applied to classify all subsequent traffic data accordingly. The experimental results are shown in Figure 4(a).

The experimental results indicate that using the static threshold method for coarse-grained long- and short-flow classification leads to inconsistent performance. The classification accuracy fluctuates, sometimes achieving high performance and other times low performance, with no significant overall improvement.

Single dataset dynamic long- and short-flow classification. To adapt to the evolving and complex nature of network traffic, this article introduces a dynamic long- and short-flow classification model update algorithm for a single dataset. Periodically, a new model is trained based on the latest traffic data, which varies over time. Consequently, each trained model differs, tailored to classify traffic specific to its corresponding period. The experimental results are presented in Figure 4(b). Experimental results show that when only the most recent data are used for training coarse-grained long- and short-flow classification at regular intervals, the performance metrics of the classification remain suboptimal, show little improvement, and may even deteriorate.

Fusion dataset dynamic long- and short-flow classification based on KDN-FLB. To improve the accuracy and stability of dynamic coarse-grained long- and short-flow classification, this article designs a dynamic flow classification model update scheme based on KDN-FLB using a fused dataset. The specific process is as follows:

- Clients request participation in the training process and prepare their local flow data.
- In each period, clients in FL train local models on the client data, generate local models, and calculate the respective flow classification thresholds.
- Under the coordination of the control plane, the local models from different clients are aggregated to update the global model and the flow classification thresholds. This ensures the accuracy and adaptability of the model.
- The new global model and flow classification thresholds are combined with the previous global model and thresholds to generate the final global model and flow classification thresholds.
- Blockchain is used to record the final global model and flow classification thresholds to prevent tampering and ensure traceability. Only authorized users can obtain the global model.
- Users utilize the newly obtained global model to classify flow data into long and short flows. The classification results are then provided to the scheduling module to improve traffic scheduling, reduce packet loss, and enhance transmission stability.

The results of the fusion dataset dynamic long- and short-flow classification based on KDN-FLB are illustrated in Figure 4(c). The experimental findings show that periodic coarse-grained long- and short-flow classification training, which incorporates both previous and current flow data, leads to a steady improvement in classification performance. Ultimately, the performance stabilizes at more than 99%, indicating a highly favorable outcome.

Dynamic thresholds. As network traffic dynamically changes, the threshold for classifying traffic into long and short flows varies accordingly. This article illustrates the dynamic threshold changes for traffic classification as shown in Figure 4(d).

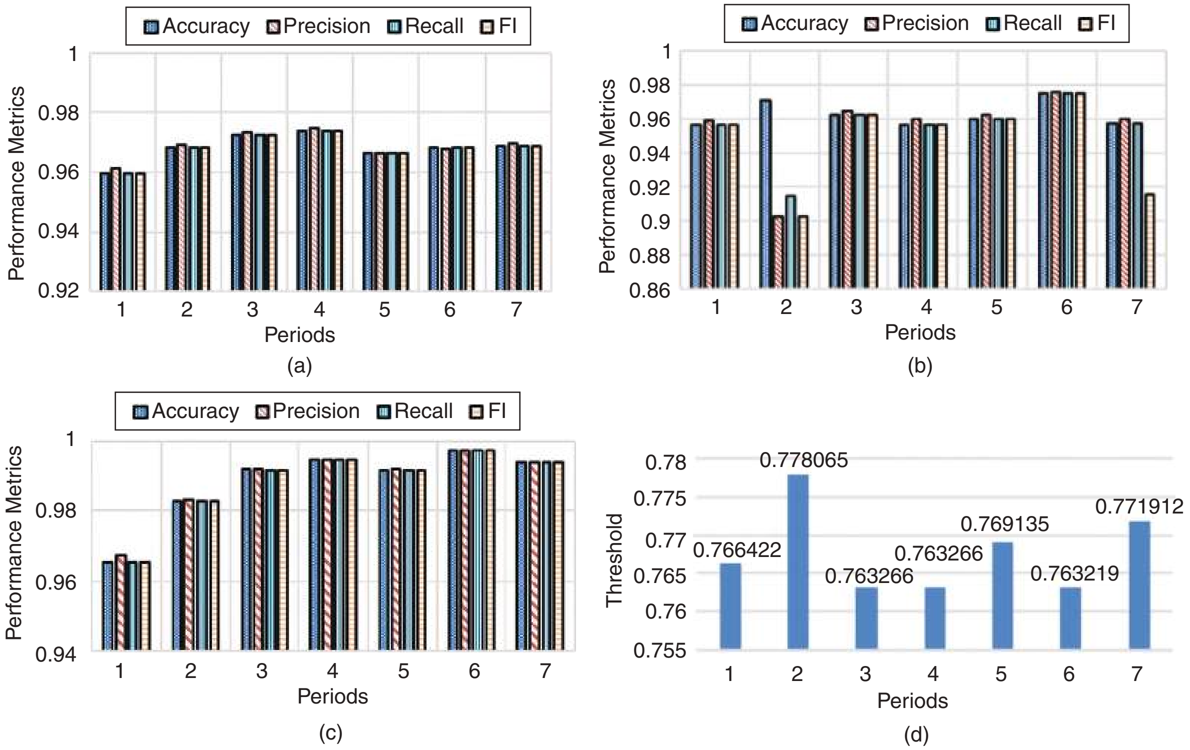


FIGURE 4. The results for the (a) static model classification scheme, (b) single dataset dynamic classification scheme, (c) fusion dataset dynamic classification scheme, and (d) dynamic flow size threshold.

CHALLENGES AND DISCUSSION

KDN-FLB presents a promising approach to enhancing security and empowering the network with self-learning, self-adaptation, and self-adjustment capabilities. Nevertheless, it has its own set of challenges.

Scalability

In KDN-FLB, scalability challenges arise due to FL and blockchain technologies. Specifically, the growing number of clients introduces heightened communication overhead in FL and model aggregation intricacies. This challenge can be mitigated in expanded network settings by using strategies such as hierarchical FL, client selection, and model compression techniques. Scalability issues may arise with blockchain ledger growth and consensus mechanisms. To resolve this challenge, various strategies can be adopted, including sharding to facilitate parallel transaction processing, enabling off-chain transactions via state channels and sidechains,[11] managing ledger size with data pruning, and integrating cross-chain technologies. Addressing these scalability challenges allows for a more adept design and implementation of KDN-FLB, ensuring high scalability and efficiency in large-scale network environments.

Energy consumption

In the KDN-FLB architecture, FL poses a risk of increased energy consumption, especially due to training models on resource-constrained devices. Additionally, integrating consensus algorithms into the blockchain raises energy usage concerns. It is essential to integrate optimized FL model training, energy-efficient blockchain consensus mechanisms, adaptive energy management,[12] renewable energy sources,[13] and energy sharing into the KDN-FLB system to ensure its sustainability and effectiveness.

Network latency

Communication and coordination between FL and blockchain are optimized and managed by addressing network latency in KDN-FLB. It is crucial to adopt strategies such as using latency-optimized FL algorithms[14] integrating distributed edge intelligence, optimizing blockchain networks through efficient consensus mechanisms, deploying adaptive asynchronous mechanisms,[15] and utilizing hybrid blockchain models. These measures reduce network latency's adverse effects on KDN-FLB architecture performance.

Computational overhead

The KDN-FLB framework encounters considerable computational overhead challenges, primarily due to the demanding computational requirements of FL algorithms and the power-intensive consensus mechanisms essential for blockchain functionality. These challenges can be addressed by optimizing FL algorithms by using methods such as model pruning and knowledge distillation,[16] adopting energy-efficient blockchain consensus mechanisms,[17] utilizing hardware accelerators to boost computation efficiency, and integrating edge computing to process data closer to the source. The KDN-FLB framework benefits from these targeted interventions by enhancing network intelligence and security under decentralized circumstances.

Interoperability

In the KDN-FLB architecture, addressing interoperability challenges in integrating blockchain platforms and FL becomes imperative. Developing communication and data exchange standards may be crucial to ensuring the seamless integration of the two technologies. The KDN-FLB architecture encompasses establishing universal standards,[18] designing cross-platform communication application programming interfaces, and fostering consortium and collaborative efforts for standardization to ensure interoperability issues.

Deployment of the KDN-FLB in real-world environments

The deployment of the KDN-FLB framework in real-world networks presents several challenges, including guaranteeing technical compatibility across various hardware and software ecosystems, overcoming bandwidth and computational resource limitations, and navigating cross-domain collaboration. Addressing these challenges necessitates a multifaceted approach that includes adapting the framework to be modular and flexible,[19] harnessing advanced technologies like 5G/6G and edge computing to mitigate resource constraints,[20] and establishing robust governance models that facilitate trust and cooperation among stakeholders while ensuring data privacy and integrity. Furthermore, continuous engagement with stakeholders and creating a feedback loop are crucial for the iterative refinement of the framework, ensuring its effectiveness and relevance. With these solutions, the KDN-FLB framework can overcome the aforementioned barriers, allowing it to significantly transform networked systems.

Considering the intricate interplay between FL and blockchain within the KDN-FLB framework is imperative for mitigating these challenges. KDN-FLB's strategic approach aims to overcome obstacles and maximize its benefits, but further research is needed.

In this article, we explore the potential of combining FL and blockchain technologies to create an intelligent KDN system, specifically known as the *KDN-FLB architecture*. The proposed KDN-FLB conceptual architecture combines the collaborative nature of FL with blockchain security and transparency features to present a decentralized and next-generation intelligent KDN architecture. The proposed KDN-FLB architecture aims to enhance dynamic and distributed network performance, enhance security measures, and empower the network with self-learning, self-adapting, and self-adjustment capabilities. We will evaluate the proposed architecture in different use cases and demonstrate its superiority to existing platforms in the future.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grants 92267206 and 62032013. Research has partially received funding from Grant Agreements 101079214 (AIoTwin), 101135576 (INTEND), and 101070186 (TEADAL) by EU Horizon. The corresponding author is Xingwei Wang.

REFERENCES

1. A. Mestres et al., "Knowledge-defined networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 47, no. 3, pp. 2–10, 2017, doi: 10.1145/3138808.3138810.
2. A. Hazra et al., "Distributed AI in zero-touch provisioning for edge networks: Challenges and research directions," *Computer*, vol. 57, no. 3, pp. 69–78, Mar. 2024, doi: 10.1109/MC.2023.3334913.
3. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, PMLR, 2017, pp. 1273–1282.
4. Y. Zhang, J. Li, Y. Yu, Z. Fan, H. Ma, and X. Wang, "SDN multi-domain routing for knowledge-defined networking," in *Proc. 15th Int. Conf. Commun. Softw. Netw. (ICCSN)*, 2023, pp. 24–29, doi: 10.1109/ICCSN57992.2023.10297317.
5. A. Rafiq et al., "Knowledge defined networks on the edge for service function chaining and reactive traffic steering," *Cluster Comput.*, vol. 26, no. 1, pp. 613–634, 2023, doi: 10.1007/s10586-022-03660-w.
6. T. A. Q. Pham, Y. Hadjadj-Aoul, and A. Outtagarts, "Deep reinforcement learning based QoS-aware routing in knowledge-defined networking," in *Proc. 14th EAI Int. Conf. Qual., Rel., Secur. Robustness Heterogeneous Syst. (Qshine)*, Ho Chi Minh City, Vietnam: Springer, 2019, pp. 14–26.
7. Q. He et al., "Routing optimization with deep reinforcement learning in knowledge defined networking," *IEEE Trans. Mobile Comput.*, vol. 23, no. 2, pp. 1444–1455, Feb. 2024, doi: 10.1109/TMC.2023.3235446.
8. Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for 5G beyond," *IEEE Netw.*, vol. 35, no. 1, pp. 219–225, Jan./Feb. 2021, doi: 10.1109/MNET.011.1900598.
9. G. Bernárdez et al., "MAGNNETO: A graph neural network-based multi-agent system for traffic engineering," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 2, pp. 494–506, Apr. 2023, doi: 10.1109/TCCN.2023.3235719.
10. M. Al-Fares et al., "Hedera: Dynamic flow scheduling for data center networks," in *Proc. 7th USENIX Conf. Netw. Syst. Des. Implementation*, San Jose, CA, USA, 2010, vol. 10, no. 8, pp. 89–92.
11. Y. Li, Y. Yu, and X. Wang, "Three-tier storage framework based on TBchain and IPFS for protecting IoT security and privacy," *ACM Trans. Internet Technol.*, vol. 23, no. 3, pp. 1–28, 2023, doi: 10.1145/3549910.
12. Y. Li et al., "Federated domain generalization: A survey," 2023, *arXiv:2306.01334*.
13. Y. Xu, Z. Liu, C. Zhang, J. Ren, Y. Zhang, and X. Shen, "Blockchain-based trustworthy energy dispatching approach for high renewable energy penetrated power systems," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 10,036–10,047, Dec. 2021, doi: 10.1109/JIOT.2021.3117924.
14. L. Toka et al., "5G on the roads: Latency-optimized federated analytics in the vehicular edge," *IEEE Access*, vol. 11, pp. 81,737–81,752, 2023, doi: 10.1109/ACCESS.2023.3301330.
15. Y. Qu, L. Gao, Y. Xiang, S. Shen, and S. Yu, "FedTwin: Blockchain-enabled adaptive asynchronous federated learning for digital twin networks," *IEEE Netw.*, vol. 36, no. 6, pp. 183–190, Nov./Dec. 2022, doi: 10.1109/MNET.105.2100620.
16. Y. Jiang et al., "Model pruning enables efficient federated learning on edge devices," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10,374–10,386, Dec. 2023, doi: 10.1109/TNNLS.2022.3166101.
17. N. Lasla, L. Al-Sahan, M. Abdallah, and M. Younis, "Green-PoW: An energy-efficient blockchain proof-of-work consensus algorithm," *Comput. Netw.*, vol. 214, Sep. 2022, Art. no. 109118, doi: 10.1016/j.comnet.2022.109118.

ABOUT THE AUTHORS

YING LI is pursuing a Ph.D. degree in computer science and technology at Northeastern University, Shenyang 110819, China. Her research interests include federated learning, blockchain, and edge intelligence. Li received an M.S. in computer technology from Northeastern University, Shenyang, China, in 2020. She is a Student Member of IEEE. Contact her at liying1771@163.com.

PRAVEEN KUMAR DONTA is a senior lecturer at Stockholm University, 16425 Stockholm, Sweden. His research interests include learning-driven distributed computing continuum systems, the cyberphysical continuum, and intelligent data protocols. Donta received a Ph.D. from the Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, India. He is a Senior Member of IEEE. Contact him at praveen@dsv.su.se.

XINGWEI WANG is a professor at the College of Computer Science and Engineering, Northeastern University, Shenyang 110819, China. His research interests include cloud computing and the future Internet. Wang received a Ph.D. in computer science from Northeastern University, Shenyang, China, in 1998. He has published more than 100 journal articles, books, and book chapters and refereed conference papers. He is a Member of IEEE. Contact him at wangxw@mail.neu.edu.cn.

ILIR MURTURI is a postdoctoral researcher at the Distributed Systems Group, TU Wien, 1040 Wien, Austria. His current research interests include the Internet of Things, distributed computing continuum systems, edge AI, and privacy in distributed, self-adaptive, and cyberphysical systems. Murturi received a Ph.D. from the Distributed Systems Group, TU Wien, Vienna, Austria. He is a Member of IEEE. Contact him at imurturi@dsg.tuwien.ac.at.

MIN HUANG is currently a professor at the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China. Her research interests include modeling and optimization for logistics and supply chain systems. Huang received a Ph.D. in control theory from Northeastern University, Shenyang, China, in 1999. She has published more than 100 journal articles, books, and refereed conference papers. She is a Member of IEEE. Contact her at mhuang@mail.neu.edu.cn.

SCHAHRAM DUSTDAR is a full professor of computer science (informatics) with a focus on internet technologies heading the Distributed Systems Group at TU Wien, 1040 Wien, Austria. His research interests include edge-computing, fog-computing, cloud computing, and human-based computing as well as AI in the co-evolution of distributed systems. He is an associate editor of *IEEE Transactions on Services Computing, ACM Transactions on the Web*, and *ACM Transactions on Internet Technology* and is on the editorial board of *IEEE Internet Computing* and *Computer*. He is the editor-in-chief of *Computing* (an SCI-ranked journal of Springer). He is a Fellow of IEEE and a member of the Academia Europaea. Contact him at dustdar@dsg.tuwien.ac.at.

18. S. Schindler and S. Marvin, “Constructing a universal logic of urban control? International standards for city data, management, and interoperability,” *City*, vol. 22, no. 2, pp. 298–307, 2018, doi: 10.1080/13604813.2018.1451021.

19. S. Otoum, I. A. Ridhawi, and H. Mouftah, “A federated learning and blockchain-enabled sustainable energy trade at the edge: A framework for industry 4.0,” *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3018–3026, Feb. 2023, doi: 10.1109/JIOT.2022.3140430.

20. G. Qu, N. Cui, H. Wu, R. Li, and Y. Ding, “ChainFL: A simulation platform for joint federated learning and blockchain in edge/cloud computing environments,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3572–3581, May 2022, doi: 10.1109/TII.2021.3117481.

A Flexible Software-Defined Networking-Based Privacy-Preserving Method for Internet of Things-Based Smart City Environment Based on the Neighbors Situation

Mehdi Gheisari, Shaoxing University and Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences
Hamid Tahaei, Shaoxing University
Christian Fernández-Campusano, University of Santiago of Chile
Mazhar Malik, University of the West of England
Ernest Mnkandla, **Zenghui Wang**, **and Malusi Sibiya**, University of South Africa
Julian L. Webber, Kuwait College of Science and Technology
Muhammad Rizwan Mughal, Sultan Qaboos University
Cheng-Chi Lee, Fu Jen Catholic University and Asia University
Panjun Sun, Shaoxing University
Abolfazl Mehbodniya, Kuwait College of Science and Technology

We introduce "DPSmartCity," a context-aware dynamic software-defined networking (SDN) framework that preserves privacy in smart cities. Enhancing the Internet of Things (IoT)-centric infrastructure with dynamic network management, the SDN controller employs contextual information and trust evaluation, determining secure routes from IoT devices to the cloud.

Digital Object Identifier 10.1109/MC.2024.3506700
Date of current version: 28 April 2025

Due to the anticipated surge in the global population by 2050, there is a critical need to put forth strategies to tackle urban challenges, such as enhancing transportation and accessibility, improving social services, promoting sustainability, and empowering citizen involvement.[1]

INTRODUCTION

Two transformative technologies with the potential to significantly impact urban management are the Internet of Things (IoT) and cloud computing. By integrating IoT with cloud computing, cities can tackle challenges more effectively, leading to the development of what is known as an *IoT-based smart city*. An IoT-based smart city comprises various components, such as smart lighting systems, autonomous delivery networks, and flying ad hoc networks (FANETs). IoT devices in smart cities, whether mobile, such as those in FANETs, or stationary, like fire hydrants, exchange data with cloud computing resources to facilitate better decision making and support complex analytics. Figure 1 provides an overview of key IoT application domains.[2,3]

In the context of data abundance, while it provides opportunities for enhanced services, IoT poses a significant privacy concern. IoT devices within smart cities may generate sensitive data, such as headcounts in crucial locations. Inadvertent disclosure of these data could render the system susceptible to malicious activities, potentially resulting in digital or physical harm. Therefore, stringent measures must be in place to prevent unauthorized data exposure or maintain privacy integrity during collaborative efforts. Even the smallest weakness in privacy breach can be taken advantage of, showing that the strength of smart cities depends on their most vulnerable areas. By effectively addressing the privacy challenge, which is the aim of this article, citizens will be more inclined to trust and engage with the IoT-driven smart city infrastructure. Through proactive measures and vigilant leadership, cities can minimize the risk of breaches.

Upon examination, it is evident that current solutions fall short in delivering a satisfactory environment. They tend to be either expensive or lacking in effectiveness concerning privacy protection, as indicated by recent research findings.[4] Striking a balance between cost and privacy preservation poses a significant obstacle to achieving an effective smart city infrastructure.

On the other hand, a software-defined networking (SDN) paradigm has appeared, which separates the data plane from the control plane. This separation leads to having a network that is manageable in a flexible manner. It offers a centralized view of an entire network that makes it easier to streamline the network management and provisioning. Moreover, it supports the control of the network equipment from a single centralized controller, which cannot be done using the "simple network management protocol." Furthermore, in the SDN environment, controlling data traffic is one of the main advantages. By implementing SDN, we can change the network configurations in a convenient manner. We will leverage this advantage for our solution.

Technically, our solution follows the following steps in detail.

1. We equip the current smart cities with an SDN paradigm to manage the IoT-based smart city, including data flow, and maintain its flexibility.
2. We propose a solution on top of the equipped environment to preserve IoT devices' privacy. In the case that an IoT device does not produce sensitive data, such as temperature, it sends its data to the SDN controller via Dijkstra. However, if the IoT device produces sensitive data, the SDN controller commands the corresponding IoT device to reroute its data if its trust in its neighbor

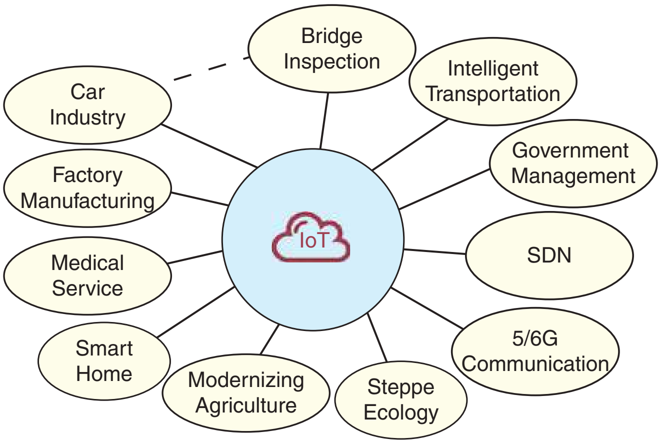


FIGURE 1. Overview of IoT applications.

is lower than 50%, and it selects the nearest neighbor, which has trust more than 50%, as the forwarding node. Precisely, when the IoT device in the smart city produces sensitive data, the SDN controller manages the routing of the IoT device data. If the amount of trust of the IoT device in its neighbor is lower than 50%, the SDN controller does not allow the data to be routed from the predefined route and should be flowed from a new route through the closest neighbor in which it has the trust of more than 50%. However, if the trust amount exceeds 50%, the sensitive data are passed from the predefined path. On the other hand, if the produced data are not sensitive, the IoT device sends its data via the SDN controller's predefined route. Finally, the SDN controller sends the received data to the cloud computing space for further analysis in both cases.

3. Finally, we evaluate the proposed method from various evaluation metrics, including the amount of imposed overload, privacy-preserving degree (how much time adversaries need to perturb the generated sensitive data), and latency. Moreover, we compare our solution with several well-known current studies to show its superior performance.

The article is organized as follows. The "Related Work" section describes related work and literature. Then, we explain the proposed solution in detail in the "Proposed Solution" section. The "Performance Evaluation" section evaluates and compares the performance of our proposed solution in this article. Finally, in the last section we conclude the article and suggest some future work that can be done to have a more efficient IoT-based smart city.

RELATED WORK

In this section, we focus on the state of the art to find what has been done in the academic world and the drawbacks for addressing our research problem.

Sahil et al.[5] proposed a framework to provide end-to-end security and privacy in vehicular networks enabled by 5G. Their proposed framework simplified network management by leveraging the SDN paradigm while obtaining optimized network communication. It includes two modules. The first module is an authentication protocol that leverages an elliptic curve cryptography for mutual authentication of cluster heads and certificate authority using the SDN in vehicular environments. The second designed module is an intrusion detection module to detect the system's potential intrusions. To harness the potential advantages of the proposed framework, they leveraged three simulators (for example, NS3, SUMO, and SPAN). They evaluated the first module based on the security features, while the second one was based on the detection rate, false-positive rate, accuracy, detection time, and communication overhead. They compared the second module with state of the art. They also showed that their solution has low computational complexity. Although their solution increases the security level of the autonomous vehicle environment, they did not calculate how much their solution can preserve the autonomous vehicle's privacy.

Gheisari et al.[6] designed a method for preserving the privacy of IoT devices in a smart city using the SDN paradigm. Their solution is context-aware, which denotes that their solution can react to the environment based on context. At first they equipped the smart city with the SDN paradigm. Then, they mounted a privacy-preserving method so that if the device produces sensitive data, it divides its data into two parts, 70% and 30%, in the case that the first division is passing through the most known secure path to the SDN controller, and the remaining portion is from a created virtual private network (VPN) to the SDN controller. As the next step, the controller aggregates received data from the device. Finally, it sends the aggregated data to the cloud computing space for further analysis and to get commands from it. They evaluated their proposed method via several evaluation metrics, like accuracy, penetration time, and overload. Furthermore, they compared their solution with the state of the art. Although their solution poses more overload on the smart city, it resists against unintentional disclosure of sensitive data more efficiently. However, their solution does not consider an adversarious situation in the network.

Although Lu et al.[7] did not leverage the SDN paradigm, we were inspired by it. They proposed a lightweight privacy-preserving data aggregation (LPDA) method that is lightweight in the fog-computing–facilitated IoT environment. They leveraged the homomorphic Paillier encryption method to encrypt the flowing data. Further, they applied the Chinese remainder theorem for hybrid data aggregation of collected data from diverse IoT devices. In addition, to provide a more efficient solution, they used a one-way hash chain function to be able to filter injected false data at the network edge-level forging to have more efficient authentication of IoT devices. As a supplementary step, they leveraged a differential privacy-preserving technique to be able to achieve a better privacy-preserving degree. Their evaluation results show that their solution increases the amount of privacy-preserving degrees. Moreover, their solution is lightweight, so that it can be applied to real-time demanding environments. Beyond the mentioned advantages, LPDA has a drawback: It is not flexible and agile.

Tabassum and Ibrahim[8] proposed a model for the smart city that ensures the users' privacy and integrity of services. It avoids misuse of public data by malicious service providers. It implements end-to-end security and privacy to guarantee secure services in smart cities. In this regard, they proposed a hybrid strategy by dividing the security protection at the macro level. It gives the capability of separating protection-based portions within the smart city systems while being a traceable solution. Regretfully, the authors have not evaluated their solution to find how much it can prevent unintentional disclosure of sensitive data.

PROPOSED SOLUTION

Having mentioned the previous research, our solution aims to provide a better privacy-preservation amount for IoT devices in the smart city than current studies. In this section, we explain our solution in detail.

Assumptions

In our IoT setup, each device knows the level of trust it has with its neighbors, stored in a database with three columns: IoT-Device ID, neighbor ID, and trust level. The trust values range from 0 to 100, based on a random number. The structure of the IoT device network remains constant in our smart city model, as detailed in Gheisari et al.[9] Moving forward, there are unresolved queries on how nodes decide whom to trust, how much, and the dynamics of gaining or losing trust. While these unresolved inquiries are beyond the scope of this article, further exploration is warranted to address these inquiries.

Procedure

Our solution carries the idea of preserving the IoT devices' privacy based on the trust amount between the nodes dynamically. Specifically, we facilitate the IoT-based smart city environment with the SDN paradigm as the first step. Then, on top of the equipped smart city, we mount a method for privacy-preservation of IoT devices that produce sensitive data, the data whose disclosure may cause harm to the system. In this way, IoT devices behave smarter based on the context to prevent the privacy breach. Specifically, if an IoT device finds it does not have enough trust in its neighbor (less than 50%), it would not send its sensitive to the SDN controller via it. In this case, it sends its data through a more trusted neighbor. In a rare case, if it could not find a neighbor with enough trust, it will create a VPN and send its data to the SDN controller. However, when the amount of trust is more than 50%, the data holder transfers its data to its neighbor that has trust ≥50%. In turn, the IoT device forwards its data to the new neighbor. Finally, the SDN controller sends the received data to the cloud computing for further analysis.

The 50% threshold was selected as a heuristic to represent a balance point between trust and suspicion. While the chosen threshold is not definitive, the threshold can be adjusted or even modeled probabilistically to account for varying degrees of trustworthiness based on the specific security and operational requirements of the IoT network.

We leverage the OpenFlow network in which the first packet of each network traffic flow is forwarded to the central controller through a secure channel known as *the southbound interface*.[10] In addition, network applications and applicaton programming interfaces can be installed on top of the central controller and accessed via a *northbound interface*. When a flow originating from an end-device (that is, IoT device) reaches a switch, the first packet of that flow is dispatched to the controller. This packet is termed a *Packet-In* message and typically includes details about the flow. Upon receiving the Packet-In message, the controller analyzes the information to determine the optimal route for forwarding the flow throughout the network. It then sends instructions to the SDN forwarding element via a *Packet-Out* message. This message contains the entire flow's details, which are subsequently populated into the flow table(s) of the forwarding element.

An OpenFlow switch consists of various flow properties, including match fields, priority, counters, instructions, timeouts, cookies, and flags. These properties provide instructions on how incoming packets should be processed and forwarded based on their characteristics, such as source and destination addresses, ports, and protocol types. Each entry in a flow table is distinguished by its match fields and priority level. These components establish a distinct flow entry within a particular flow table. Within each entry, a series of instructions is defined. The instructions imply the actions to be taken when the remaining packets of a flow aligns with the entry's criteria. These actions may involve altering packet attributes, modifying the action set, or directing the packet through specific stages of the processing pipeline. Furthermore, counters are associated with each flow entry that track various metrics related to the flow, such as flow duration, the port through which it traverses, and any associated group memberships, etc.

Based on the OpenFlow specification, the central controller has the capability to assign a “priority” attribute to each flow, allowing it to dictate which flow entry should be selected based on policies defined by the controller. Essentially, only the highest priority flow entry that matches the packet’s criteria will be chosen. This enables the OpenFlow forwarding elements to prioritize one entry among several possible matches. This property proves beneficial in scenarios, such as traffic engineering, load balancing, and security enforcement. While this prioritization mechanism addresses various shortcomings, we argue that routing tables could benefit from a more specific attribute to ensure a dynamically privacy-preserving routing mechanism; such an attribute would enhance the protection of sensitive information during packet forwarding, thereby bolstering privacy and security measures in network communication.

As an example, assume an IoT device in an IoT- based smart city wants to send its data to cloud computing space through an SDN controller via several middle nodes. Unfortunately, several middle nodes are adversaries, and they have bad intentions (the layout has been mentioned in Gheisari et al.[9]); in our scenario, they are “honest but curious” about the flowing sensitive data. However, they should not be able to find it and breach IoT devices’ privacy; otherwise, they may disclose it to a third party/parties, forging harm to the system. Our solution prevents this unintentional disclosure.

We leverage the database used in Kannan et al.[11] with a similar setup in MININET Wi-Fi. In this scenario, several IoT devices have trust values assigned to their neighboring devices, selected randomly at the start. For instance, Camera 1 is linked to Camera 3 with a trust amount of 24, indicating a level of trust or interaction between the two. Camera 1 is also connected to Camera 4 and Camera 5, with trust values of 69 and 54, respectively, reflecting varying degrees of trust within the network. Likewise, Camera 2 is connected to Camera 4 and Camera 5 with trust values of 41 and 61, respectively, forming a network of interconnected devices with established trust relationships.

Additionally, devices like Color 1 and Color 2 are interconnected with other devices, such as Tilt device and Speed 1, each having distinct trust amounts denoting the reliability or credibility of these connections.

Algorithm 1 shows the procedure of the proposed solution in this research.

As a conclusion for the proposed architecture explained above, by leveraging trust-based routing and encryption, the proposed solution ensures that even in the presence of curious adversaries, the sensitive data remain secure. The system efficiently balances privacy preservation with minimal computational overhead; it also ensures that the latency remains low and computational costs are within limits. When compared to traditional methods, this solution demonstrates better performance in protecting IoT devices’ privacy while maintaining efficient data transmission.

In real-world implementations, trust scores are generally determined using a combination of direct and indirect interactions between devices. Some of the commonly used methods are: historical interactions (that is, the past interactions between devices); behavioral analysis, where trust can be inferred by monitoring the behavior of devices; and reputation systems, where trust scores can also be calculated using feedback from other devices in the network.

ALGORITHM 1: PROPOSED PRIVACY-PRESERVING PSEUDOCODE

```
Input: $X$ = The sensed data, $Y$ = IoT device ID
for all IoT devices do
   if $X$ = Sensitive data then
      The SDN controller specifies a route from $Y$ to itself
      $T$ = Next Neighbor Trust
      $NID$ = Next Neighbor ID
      if $T \leq 50$ then
         $NID$ = Nearest neighbor ID ($NID$) with $T$ more than 50
         if $NID$ = Empty then
            Create a VPN from the device and send to the SDN controller
            else
            The data holder forwards its data to the $NID$
      The SDN controller sends the obtained data to the Cloud Computing
       space
Output: NULL
```

PERFORMANCE EVALUATION

This section evaluates our proposed method using several metrics and compares it with well-known current studies to assess its effectiveness. We evaluate the method in terms of computational cost, privacy-preserving capability, and latency. In our scenario, devices are positioned within a 10 × 10 m^2 area, with a distance of 0.1 × 0.1 m^2 between them.

Computational cost

This section evaluates the proposed solution by examining the computational overhead it imposes on the system. We then compare the proposed solution with several well-known studies to highlight its superior performance.

Figure 2 depicts that our method imposes 30% computational overhead on the system in the first 8 s; afterward, it suffers less overhead, as shown in the figure. We may deduce from the plot that the system has shifted from a transient or initial state (where conditions are changing rapidly) to a steady state (where conditions have stabilized) after 8 s. To validate the performance of our model, we have compared the proposed model with two well-known studies called *CS*[6] and *MPIoT-SDN*.[12] In the CS approach, the authors, at first, equipped the IoT-based smart city environment with the SDN paradigm. They then proposed a privacy-preserving approach on top of it, whereas if an IoT device senses sensitive data, it splits its data into two parts, 70% and 30%. The IoT device sends its first part to the SDN controller through the most secure route defined by the SDN controller and the second part (the remaining part) through a created VPN by the SDN controller to itself. On the other hand, in the MPIoT-SDN work, the authors first equipped the IoT-based smart city with the SDN paradigm. Then, the SDN controller divides the IoT devices into two categories using the K-nearest neighbors method, a well-known machine-learning algorithm used for classification tasks. If a device is located in the first category, the SDN controller specifies two different routes, and the IoT device halves its data as well. Next, the IoT device sends its first half through the first route and the second half through the second route. Finally, the SDN controller integrates the obtained divided data and sends it to the cloud computing space for further analysis. This traffic-splitting mechanism is implemented as an additional privacy-preserving measure. By dividing the data into two parts and sending each half through a different route, the system makes it significantly harder for adversaries to intercept and reconstruct the complete data. However, if the IoT device locates in the second category, it encrypts its data based on the SDN controller command and sends it to the SDN controller, and consequently to the cloud computing space.

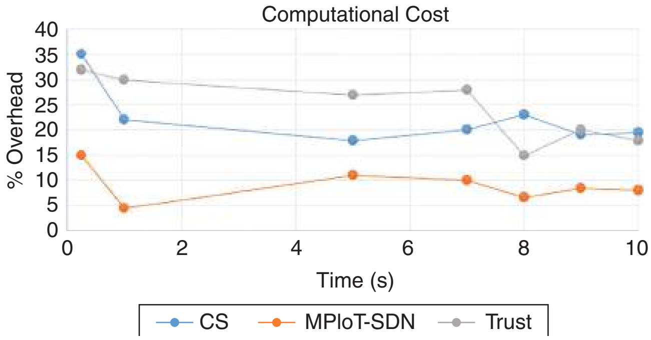


FIGURE 2. Computational overhead.

Although our solution always imposes more overhead than the MPIoT-SDN method, we need to consider its performance through other evaluation metrics, such as privacy-preserving degrees and latency, evaluated in the following sections.

Penetration rate

This section calculates how much our solution can prevent unintentional disclosure of IoT devices' sensitive data against adversaries. Moreover, we compare it with several well-known current studies. We consider the same threat model and setting as introduced in Gheisari et al.[6] Briefly, as a threat model, there are two "honest but curious" adversaries trying to eavesdrop on the flowing sensitive data, which in our scenario are Camera 2 and Speed 1. Speed 1 only eavesdrops on the flowing sensitive data, whereas Camera 2 has background information about the ecology procedure, how sensitive data flow, and which routes they pass. In our scenario, the Speed 1 device has been invaded by "simple attack" while the Camera 2 has been invaded by a "linked data attack." The aim is to prevent this unintentional disclosure of data. We calculate the time these two adversaries take to find the originated sensitive data. Then, we also compare it with the CS method that has been introduced in Gheisari et al.[6] and described in the related work.

Figure 3 illustrates the evaluation of our solution and compares it with well-known current studies on how much it can prevent the unintentional disclosure of data.

As Figure 3 depicts, our solution cannot prevent the unintentional disclosure of sensitive data against the "simple attack" as effectively as the CS method. The CS method takes 0.4 s, while our solution takes 0.28 s. However, our solution against the "linked data attack" shows better performance than the CS approach. Our solution can resist 0.77 s against the "linked data attack," whereas the CS method takes 0.7 s.

Latency

In this section, we calculate the time required for a query to receive a response from the SDN controller.[13] Figure 4 illustrates the latency observed during 500 execution instances.

We conclude that the latency of the solution is variable and increases significantly over time, as demonstrated by the specific latency measurements after 100 and 400 execution instances.

Additional metrics

We measured several additional parameters, including the communication overhead, energy consumption, and scalability between IoT devices and the SDN controller. The communication overhead in our simulations remained below 10%. The energy consumption of IoT devices was assessed, with emphasis on how trust-based routing affects battery life. Our analysis suggests that the energy consumption per device increases due to the dynamic trust evaluation but remains within acceptable limits. The scalability is evaluated by increasing the number of IoT devices and SDN controllers in the network. Being lightweight, our method scales well up to a large number of devices with minimal impact on latency and computational overhead.

SDN controller replacement

We have investigated a dynamic SDN controller replacement mechanism to enhance system reliability and mitigate potential bottlenecks. The mechanism dynamically identifies when an SDN controller is overloaded or becomes a bottleneck. As part of the mechanism, when the load exceeds a predefined threshold (for example, 80%), the system triggers a replacement procedure. The procedure involves distributing the load to neighboring SDN controllers to ensure continuity of service. The impact of the SDN replacement mechanism for fault mitigation and system performance shall be evaluated in our future work.

Limitations

Although we have evaluated our solution from different evaluation metrics, it carries some limitations, such as:

- The initial threshold for trust is based on an assumption

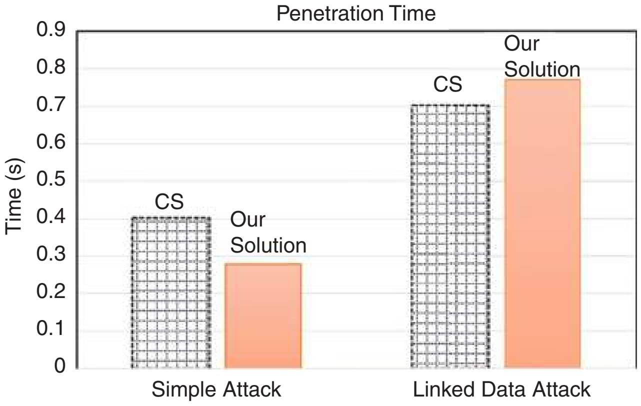


FIGURE 3. Prevention amount against unintentional sensitive data disclosure.

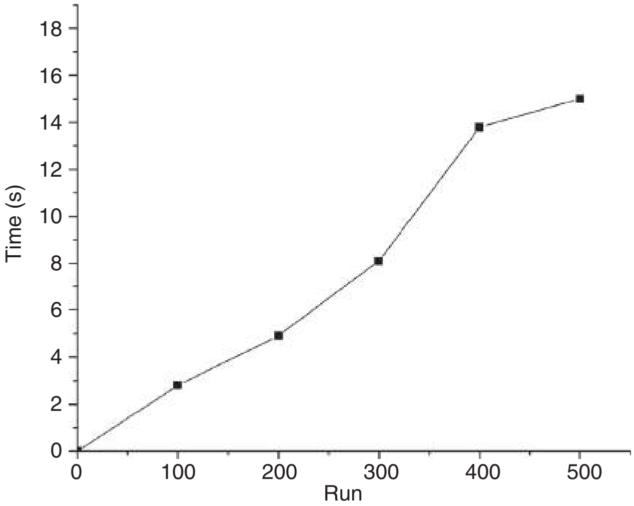


FIGURE 4. Latency amount.

ABOUT THE AUTHORS

MEHDI GHEISARI is with the Institute of Artificial Intelligence, Shaoxing University, Zhejiang 312000, China, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, India, and with the Department of Computer Science, Islamic Azad University, Shiraz branch, Iran. His research interests include information security, artificial intelligence, the Internet of Things, smart city, unnamed vehicles, drones, and autonomous vehicles. Gheisari received his Ph.D. in computers from Guangzhou University. Contact him at mehdi.gheisari61@gmail.com.

HAMID TAHAEI is a faculty member at the AI Institute, Shaoxing University, Zhejiang 312000, China. His research interests include information security, the Internet of Things, and software defined networking. Tahaei received his Ph.D. in computer science from the University of Malay. He is a Member of IEEE. Contact him at tahaei.hamid@usx.edu.cn.

CHRISTIAN FERNÁNDEZ-CAMPUSANO is a civil engineer at the University of Santiago de Chile, 9170124 Santiago, Chile. Currently, he is part of the Telecommunications Area of the Department of Electrical Engineering, where he works mainly with reliable telecommunications and computer systems. His research interests include dependable distributed and ubiquitous computing, security, and safety in ubiquitous systems, cybersecurity, Big-Data/AI, wearable and IoT networking. Fernández-Campusano received his Ph.D. in computer science from the University of Basque Country in Spain. Contact him at christian.fernandez@usach.cl.

MAZHAR MALIK is an associate director/head of Intelligent Systems at the School of Computing and Creative Technologies, University of the West of England, BS16 1QY Bristol, U.K. His research interests include the Internet of Things, cybersecurity, and digital forensics. Malik received his Ph.D. in computer science (cyber security), SMIEEE-100821470, from the School of Computing and Creative Technology. Contact him at mazhar.malik@uwe.ac.uk.

ERNEST MNKANDLA is a professor in artificial intelligence and software engineering at the School of Computing, University of South Africa, 0003 Pretoria, South Africa. His research interests include the Internet of Things, software engineering, and ethical development of artificial intelligence technologies. Mnkandla received his Ph.D. from the University of Witwatersrand. He is a Member of IEEE. Contact him at mnkane@unisa.ac.za.

ZENGHUI WANG is a faculty member at School of Engineering, University of South Africa, Florida 1709, South Africa. His research interests include artificial intelligence, evolutionary optimization, and control systems. Wang received his doctorate from Nankai University. He is a Member of IEEE. Contact him at wangz@unisa.ac.za.

MALUSI SIBIYA is with the Centre for Augmented Intelligence and Data Science, School of Computing, University of South Africa, Pretoria 0002, South Africa. His research interests include machine learning, natural language processing, and reinforcement learning. Sibiya received his Ph.D. in science, engineering, and technology from the University of South Africa. Contact him at sibiym@unisa.ac.za.

JULIAN L. WEBBER is an associate professor with Kuwait College of Science and Technology, 13002 Safat, Kuwait. His research interests include communication engineering and telecommunications engineering. He is a Senior Member of IEEE. Contact him at j.webber@kcst.edu.kw.

MUHAMMAD RIZWAN MUGHAL is an associate professor at Sultan Qaboos University, Muscat 123, Oman, and a postdoctoral research fellow with the Department of Electronics and Nano-Engineering, Aalto University, 02150 Espoo, Finland. His research interests include application of artificial intelligence and machine learning in embedded systems, plug-and-play designs, and systems engineering. Mughual received a Ph.D. in electronics and communication engineering from the Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy, in 2014. He is a Member of IEEE. Contact him at m.mughal1@squ.edu.om.

CHENG-CHI LEE is a faculty member in the Department of Library and Information Science Fu Jen Catholic University, New Taipai City 24205, Taiwan, and the Department of Computer Science & Information Engineering, Asia University, Taichung City 413, Taiwan. His research interests include data security, cryptography, network security, mobile communications and computing, and wireless communications. Contact him at cclee@mail.fju.edu.tw.

PANJUN SUN is with the Institute of AI, Shaoxing University, Zhejiang 312000, China. His research interests include cloud computing, privacy protection, access control, trust management, and game theory. Sun received his Ph.D. in information and communication system from Shanghai Jiao Tong University. Contact him at sunpanjun2008@163.com.

ABOLFAZL MEHBODNIYA is a faculty member at School of Computing, Kuwait College of Science and Technology, 13002 Safat, Kuwait. His research interests include artificial intelligence with applications in communications engineering, Internet of Things, and real-world applications. He is a Senior Member of IEEE. Contact him at a.niya@kcst.edu.kw.

discussed earlier. This assumption could be eliminated by enhancing the solution through the use of artificial intelligence and machine learning.

- The amount of latency can be improved through extending the solution by proposing an efficient SDN controller replacement solution.
- Another limitation of the article, which needs more investigation, is the number of adversaries. In other words, our solution may not be as effective as our evaluations show when the number of adversaries varies.

This research work introduces a novel context-aware privacy-preserving method mounting on an IoT-based smart city with the SDN paradigm. Our solution preserves the IoT devices' privacy based on the amount of trust of nodes with their neighbors. We have evaluated and compared our solution with several well-known current studies. We showed that although our solution imposes more overhead than one of the investigated current studies, it can prevent unintentional disclosure of data more effectively when the adversary in some cases, had background information. In future, we plan to use machine learning to find and predict the best amount of trust. Another future work is on providing a dynamic framework so that the SDN controller makes decisions dynamically based on the trust of neighbors.

ACKNOWLEDGMENT

Special thanks to Islamic Azad Unviersity, Iran, with the grant number of 1337132871361. The corresponding authors for this article are Panjun Sun, Cheng-Chi Lee, and Mehdi Gheisari.

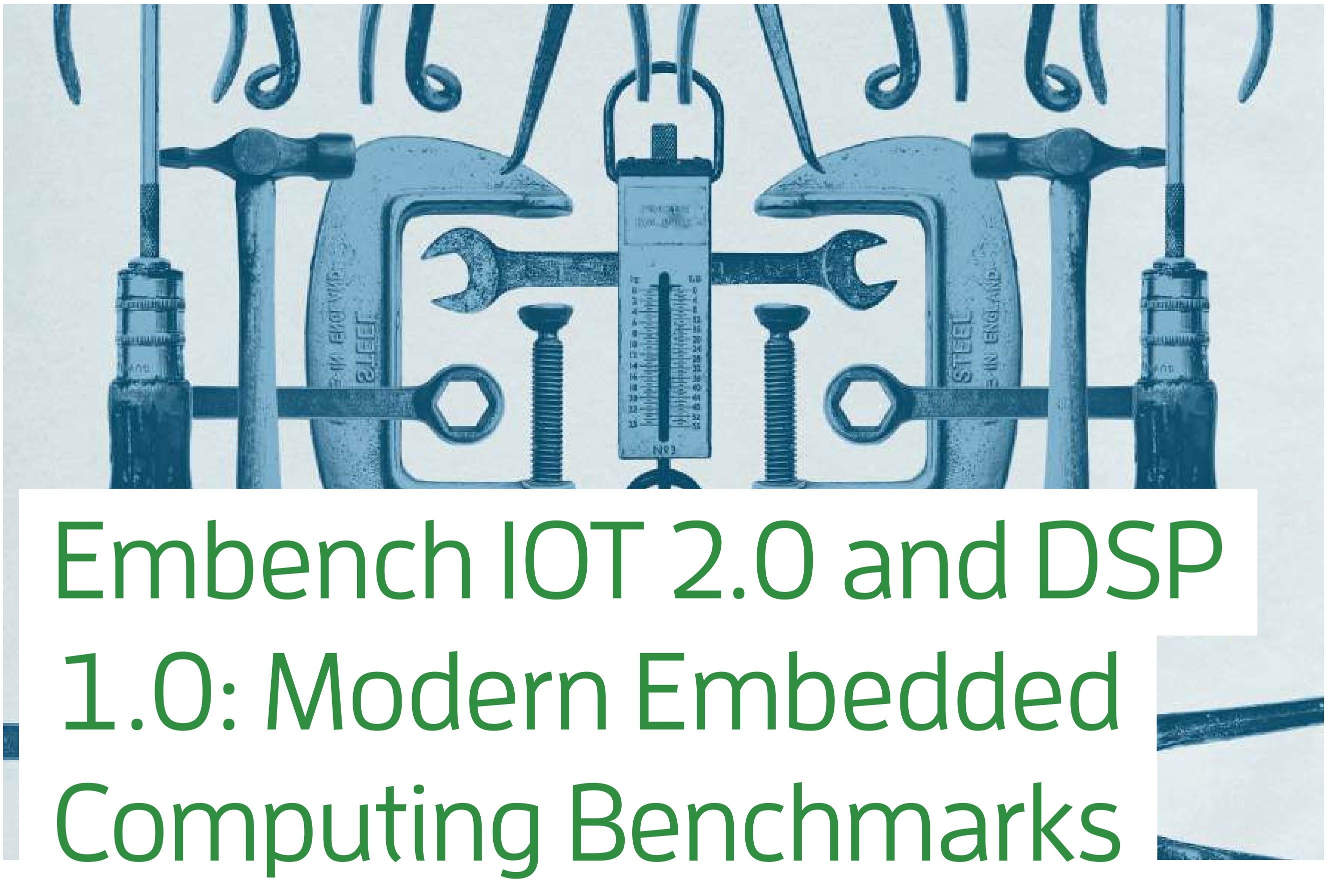
REFERENCES

1. H. Elahi, G. Wang, W. Jiang, A. Bartel, and Y. Le Traon, "A qualitative study of app acquisition and management," *IEEE Trans. Comput. Social Syst.*, vol. 11, no. 2, pp. 1907–1925, Apr. 2024, doi: 10.1109/TCSS.2023.3288562.
2. P. Sun, Y. Wan, Z. Wu, and Z. Fang, "A survey on security issues in IoT operating systems," *J. Netw. Comput. Appl.*, vol. 231, Nov. 2024, Art. no. 103976, doi: 10.1016/j.jnca.2024.103976.
3. P. Sun, S. Shen, Y. Wan, Z. Wu, Z. Fang, and X-Z Gao, "A survey of IoT privacy security: Architecture, technology, challenges, and trends," *IEEE Internet Things J.*, vol. 11, no. 21, pp. 34,567–34,591, Nov. 2024, doi: 10.1109/JIOT.2024.3372518.
4. N. Waheed, X. He, M. Ikram, M. Usman, S. S. Hashmi, and M. Usman, "Security and privacy in IoT using machine learning and blockchain," *ACM Comput. Surv.*, vol. 53, no. 6, pp. 1–37, 2021, doi: 10.1145/3417987.
5. S. Garg, K. Kaur, G. Kaddoum, S. H. Ahmed, and D. N. K. Jayakody, "SDN-based secure and privacy-preserving scheme for vehicular networks: A 5G perspective," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8421–8434, Sep. 2019, doi: 10.1109/TVT.2019.2917776.
6. M. Gheisari, G. Wang, W. Z. Khan, and C. Fernández-Campusano, "A context-aware privacy-preserving method for IoT-based smart city

using software defined networking," *Comput. Secur.*, vol. 87, Nov. 2019, Art. no. 101470, doi: 10.1016/j.cose.2019.02.006.

7. R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing -enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017, doi: 10.1109/ACCESS.2017.2677520.
8. D. Pradhan, M. Behera, and M. Gheisari, "Dynamic data placement strategy with network security issues in distributed cloud environment for medical issues: An overview." *Recent Adv. Comput. Sci. Commun. (Formerly: Recent Pat. Comput. Sci.)*, vol. 17, no. 6, pp. 25–38, 2024.
9. M. Gheisari et al., "An agile privacy -preservation solution for IoT -based smart city using different distributions," *IEEE Open J. Veh. Technol.*, vol. 4, pp. 356–362, 2023, doi: 10.1109/OJVT.2023.3243226.
10. P. Göransson, C. Black, and T. Culver, "The OpenFlow specification," in *Software Defined Networks*, Amsterdam, The Netherlands: Elsevier, 2017, pp. 89–136.
11. S. Kannan et al., "Ubiquitous vehicular ad-hoc network computing using deep neural network with IoT-based bat agents for traffic management," *Electronics*, vol. 10, no. 7, 2021, Art. no. 785, doi: 10.3390/electronics10070785.
12. M. Gheisari, G. Wang, S. Chen, and H. Ghorbani, "IoT-SDNPP: A method for privacy-preserving in smart city with software defined networking," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.*, pp. 303–312, 2018.
13. B. Montazeri, Y. Li, M. Alizadeh, and J. Ousterhout, "Homa: A receiver-driven low-latency transport protocol using network priorities," in *Proc. Conf. ACM Special Interest Group Data Commun. (SIGCOMM)*, New York, NY, USA: Association for Computing Machinery, 2018, pp. 221–235.

Digital Object Identifier 10.1109/MC.2025.3553274



Embench IOT 2.0 and DSP 1.0: Modern Embedded Computing Benchmarks

David Patterson, Google and University of California, Berkeley

Jeremy Bennett, **Mary Bennett**, and **Hélène Chelin**, Embecosm

David Harris, Harvey Mudd College

Jennifer Hellar, Cirrus Logic

William Jones, Embecosm

Konrad Moron, Technische Universität München

Paolo Savini, Embecosm

Roger Shepherd, Chipless

Ray Simar, Rice University

Zachary Susskind, University of Texas, Austin

Stefan Wallentowitz, Hochschule München

This article introduces two modern benchmarks, Embench IOT 2.0 and DSP 1.0. Embench 1.0 was originally inspired by the drawbacks of the two popular embedded benchmarks, CoreMark and Dhrystone. We give examples of important questions that the more precise new benchmarks can answer that the older ones cannot.

Digital Object Identifier 10.1109/MC.2024.3511352
Date of current version: 28 April 2025

Benchmarks are how vendors market computers and one way computer architects, compiler writers, and customers measure computer performance. In this article, we introduce two new benchmarks for the low end of embedded computing: Embench IOT 2.0 and Embench DSP 1.0.

Given the marketing importance of benchmarks, benchmarks shape a field for better or worse.[1] Architects identified the pitfalls of poor benchmarks in the last century,[2] leading to best practices for benchmarks today. Although the number of embedded computers is skyrocketing, this field still relies on two legacy benchmarks that fall short of best practices: CoreMark[3] and Dhrystone.[4]

To set the stage for Embench IOT 2.0 and Embench DSP 1.0, we first review CoreMark, Dhrystone, and Embench 1.0.

REVIEW OF COREMARK AND DHRYSTONE

CoreMark and Dhrystone are single *synthetic programs*, fake programs that try to match the behavior of real programs but fall far short. They dominate performance assessment of computers and compilers for embedded computing. For example:

- ARM's list of its numerous versions of Cortex-R cores always includes both CoreMark and Dhrystone performance.[5] It rarely mentions any other benchmark. In the microcontroller space, they are often the only benchmarks used.[6]
- Even high-end processor designers must aggressively tune for Dhrystone because some customers have a lower bound on acceptable Dhrystone performance to consider even high-end designs.[6]

The flaws of synthetic programs are well documented.[2] Architecture and compiler optimizations that help synthetic benchmarks are often irrelevant to real programs, and even worse, innovations that improve real programs frequently do not help synthetic benchmarks.

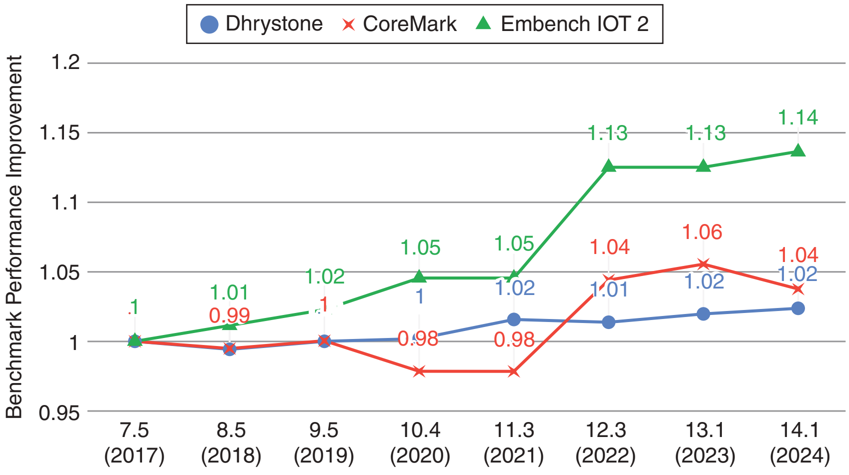


FIGURE 1. Performance for GCC versions on ARM from 2017 to 2024.

The 40-year-old Dhrystone contains unusual code that is unrepresentative of modern programs. For example, Dhrystone contains an integer division, which, in the context of a 300-instruction loop, impacts the final score. But the operands are the same for every loop iteration—it always divides 9 by 7. Some architects installed a one-entry cache inside the divider to avoid division latency for the special case of invariant arguments. Surely real programs rarely benefit from this gimmick.[6] One manufacturer even performed the division using a library function that only worked correctly on numbers in a restricted range!

Dhrystone also uses null-terminated strings that have fallen out of favor for their performance and security problems. Nevertheless, given the importance of Dhrystone, Alibaba's C-SKY has the instruction `tstnbz` (test no byte zero) that accelerates null string processing. It's a big help for Dhrystone but irrelevant for CoreMark or SPEC benchmarks.

Furthermore, library calls within Dhrystone dominate execution time. It is more benchmarking the C library optimizations for a particular platform rather than the processor.[7]

Misleading results from these two benchmarks affects compilers as well. Figure 1 measures the performance of six versions of the Gnu C Compiler (GCC) compiler since 2017. Dhrystone shows only a 2% gain in four years and nothing for the past three. CoreMark indicates performance was 2% *worse* in 2020, +6% in 2023, but dropped to +4% in 2024. In contrast, Embench IOT 2.0 shows a steady gain over time to 14%; performance never shrinks. (The "Benchmarking Code Size" and "Embench IOT 2.0 in Action" sections give evidence of the representativeness of Embench IOT 2.0.)

REVIEW OF EMBENCH 1.0

In 2019, a small group of professionals from academia and industry met to try to resolve the embedded benchmark dilemma.[8] They started with the following half-dozen best practices learned from 50 years of benchmarking:

- *Use a suite of programs*: A single program can't capture a real workload, and it is more vulnerable to targeted engineering tricks. A suite of programs (see Table 1) is likely the most important advantage over CoreMark and Dhrystone.
- *Evolve the benchmark suite over time*: A frozen benchmark is also vulnerable to focused engineering efforts, plus the mix of applications naturally changes over time. Evolution implies finding an organization to sustain a benchmark. The two legacy programs are 15+ and 40+ years old and counting, while Embench will be refreshed every few years.
- *The summary score must include* all *programs in the suite*: If not, cherry picking will lead to unrealistic claims.
- *Use geometric mean and standard deviation to summarize*: Other means can be misleading,[9] and without standard deviation we can't know if the differences are significant.[10]
- *Publish everything so that outsiders can reproduce results*: To ensure that the results are accurate and believable, we must allow others to check results.
- *Be free and easy to port*: If a benchmark is expensive or hard to run, it's unlikely to become popular.

The legacy benchmarks follow only the final best practice, which is a key reason they've endured; they may be inaccurate, but at least they are free and easy to run!

Embedded computers cover a widespread of processing power and cost. They include 8-bit to 32-bit processors that may cost one penny, and multiple high-end 64-bit chips for cars and network switches that cost US$100 each. Embench 1.0 aimed at the low end, computers whose program and data memory fit into 64 KiB: for example, IoT devices, but not smartphones or switches.

We next identified what is different about benchmarks for embedded processors versus the rest of computing. Here is our list as follows:

- *Code size matters*: The memory for the program can be a significant fraction of the cost of the embedded processor, so engineers can care as much about code size as performance. Since memory and flash storage comes in large chunks (for example, powers of 2), a 20% increase in code-size might add far more than 20% to the system cost of a product.
- *Integer only*: To get the cheapest embedded processors, many leave out hardware support for floating-point operations. Thus, the goal for Embench 1.0 was to omit floating-point intensive programs.
- *32-bit address size*: Given a 64-KiB memory, there is no need for more than 32-bit addresses.
- *Normalize performance by clock rate*: Rather than benchmark fixed processors found in laptops and servers, embedded processors are commonly synthesizable IP blocks that can be fabricated to run at many clock rates. Thus, the performance summary score is often divided by the clock rate. This metric can be misleading for larger computers since the external memory system doesn't scale with clock rate. For tiny, embedded computers, their caches have high hit ratios, and their small memory often fit on chip, so this practice is more defensible here.

Table 1 shows the suite of 19 programs we selected for Embench 1.0. They are mostly kernels and small programs that we found from prior efforts to make embedded benchmarks (see the "Related Work" section). To provide a realistic workload, we picked programs that varied in their use of three architecture features: branch intensity, memory intensity, and compute intensity.

For each of the three features, about half are medium intensity with the rest roughly evenly split between high and low intensity. Table 1 shows Embench 1.0 consisted of 13,406 lines of C code in total versus 1,890 for CoreMark and 386 for Dhrystone.

We wanted kernels that represented a wide range of use cases and were open source. Given the focus on inexpensive IoT devices, for Embench 1.0 we aimed for minimal floating point and minimal calls to libraries. (Our plan was for later versions of Embench to focus on floating point, the inspiration for Embench DSP 1.0.) Our starting point was BEEBS,[11] which in turn built on earlier open source benchmark suites.

The programs selected cover some well-known functions (for example, 32-bit cyclic redundancy check, matrix multiplication, Huffman encoding) and

TABLE 1. A suite of programs.

Embench IOT Name	Comments	Original Source	C LOC	Branching	Memory	Compute
aha-mont64	Montgomery multiplication	AHA	162	10%	1%	89%
crc32	CRC error checking 32 b	MiBench	101	14%	14%	72%
cubic*	Cubic root solver	MiBench	125	14%	16%	69%
edn	More general filter	WCET	285	10%	29%	61%
huffbench	Compress/decompress	Scott Ladd	309	23%	26%	51%
matmult-int	Integer matrix multiply	WCET	175	12%	38%	50%
minver*	Matrix inversion	WCET	187	17%	28%	55%
nbody*	Satellite N body, large data	CLBG	172	17%	10%	72%
nettle-aes	Encrypt/decrypt	Nettle	1,018	2%	20%	78%
nettle-sha256	Cryptographic hash	Nettle	349	1%	14%	84%
nsichneu	Extended Petri net	WCET	2,676	45%	54%	1%
picojpeg	JPEG	MiBench2	2,182	11%	28%	61%
qrduino	QR codes	Github	936	15%	20%	65%
sglib-combined	Simple Generic Library for C	SGLIB	1,844	26%	38%	36%
slre	Regex	SLRE	506	27%	31%	42%
st*	Statistics	WCET	117	16%	11%	72%
statemate	State machine (car window)	C-LAB	1,301	14%	72%	13%
ud	LUD composition Int	WCET	95	17%	24%	58%
wikisort	Merge sort	Github	866	20%	38%	42%
depthconv**	Depthwise convolution ML kernel from TinyML (8-bit ints)	Tensorflow Lite Micro	684	14%	19%	67%
md5sum**	Calculates the MD5 digest	GitHub	248	14%	13%	74%
tarfind**	Searches for files in tar archive	Original	121	27%	26%	46%
xgboost**	Gradient-boosted decision tree ML model for inference (8-bit ints)	XGBoost	284	15%	25%	60%

The first 19 rows show the suite of programs that make up the Embench IOT 1.0 benchmark. The four benchmarks with single asterisks were dropped from Embench IOT 1.0 for 2.0. The last four rows with double asterisks are the new programs added as part of Embench IOT 2.0. Like MiBench,[19] we evaluate each benchmark on intensity from three perspectives in the last three columns based on a dynamic analysis of the code: branching, memory (loads and stores), and compute (arithmetic and logical). The target architecture is RISC-V. Low (green) is ≤ 25th percentile, high (pink) is ≥ 75th percentile, and medium (yellow) is in between. The boundary percentiles are 13% and 19% for branches, 15% and 30% for memory, and 48% and 72% for compute. "C LOC" in the fourth column stands for the number of lines of code written in C. Embench IOT 2.0 has ~7x as much C code as CoreMark and ~35x as much as Dhrystone.

then representative application areas (for example, Advanced Encryption Standard encryption, SHA256 hashing, an n-body simulation, JPEG decoding).

Finally, we looked at the frequency of different classes of operations in the compiled code: flow-of-control, memory access, and arithmetic logic unit operations. The goal was to ensure we had some programs heavy in each class of operation, and others with a fairly even mix of these operations. Table 1 shows that we had a broad mix of types of programs.

INTRODUCING EMBENCH IOT 2.0

Given that it had been several years since we officially unveiled Embench 1.0, it was time for a revision. The goal of Embench IOT 2.0 was to cover new areas that have come to prominence, such as artificial intelligence (AI) inference. Our experience with Embench 1.0 also uncovered a few kernels that were problematic, such as—to our surprise—by heavily relying on floating-point arithmetic. Our original exclusions were based on static frequency analysis of C code to look for floating-point arithmetic, which occasionally let kernels through with high execution frequency despite few code occurrences statically. Here are the four benchmarks we dropped, as follows:

1. `nbody` invokes floating-point routines for the square root.
2. `cubic` relies on long double floating-point data types, atypical of C code and treated as 64 bits by ARM but 128 bits by RISC-V compilers.
3. `minver` heavily executes floating-point computations.
4. `st` is a statistics package that also relies heavily on floating-point calculations.

We added two machine learning (ML)/AI kernels to Embench IOT 2.0:

1. `xgboost` is an ML inference-optimized implementation of a gradient-boosted decision tree model[12] that quantizes floating-point parameters to 8-bit integers working on the MNIST handwritten digit dataset.
2. `depthconv` is a TinyML depthwise convolution kernel that operates on quantized floats, thus making it ideal for the Internet of Things (IOT) because it doesn't rely on floating-point math.

We also added two kernels representative of IOT tasks not part of Embench 1.0:

1. `tarfind` searches for files inside of a tar archive in memory. Tar is a simple-to-process format that is reasonable to use on low-resource IOT devices, so `tarfind` acts as a simple stand-in for a file system benchmark.
2. `md5sum` calculates the MD5 digest of binary messages in memory. MD5 is commonly used to verify data integrity and present in many existing codebases. MD5 is efficient enough to use for communication data on IOT devices.

INTRODUCING EMBENCH DSP 1.0

The Standard Performance Evaluation Corporation (SPEC) benchmark has had separate integer and floating-point suites for decades, so we decided to follow suit as we had originally envisioned. A natural complement to Embench IOT 2.0 is digital signal processing (DSP),[13] which normally uses floating-point representations.

Our DSP benchmark suite traces its inspiration to the original popular and widely influential DSP benchmarks suite from BDTi.[14] The BDTi DSP Kernel benchmark suite was a mix of classic DSP functions such as filters [finite-impulse response (FIR), infinite impulse response (IIR), and least mean square (LMS) filters], transforms [fast Fourier transform (FFT)], and data processing (max, control, Viterbi, and so on). When considering a new extension to Embench 1.0 to begin to cover DSP applications, we can first build on Embench IOT 2.0, which does a nice job of covering the data processing functions.

For our first DSP benchmarks we have focused on five foundational filtering and transform functions that were found in the BDTi suite:

- *Real block FIR*: FIR filter operating on a block of real data (Embench: fir_f32, config: taps256_n128)
- *Real single-sample FIR*: FIR filter that operates on a single sample of real data (Embench: fir_f32, config: taps256_n1)
- *IIR*: Infinite impulse response filter that operates on a single sample of real data (Embench: biquad_cascade_df2T_f32, config: sos3_n1)
- *FFT*: Fast Fourier transform converts a time-domain signal to the frequency domain (Embench: rfft512_f32, rfft2048_f32)
- *LMS adaptive FIR*: least-mean-square adaptive filter (Embench: lms_f32, config: taps256_n1).

We have added a few DSP benchmarks that are not in the BDTi suite that we believe are important today:

- *Real block IIR*: IIR filter operating on a block of real data (Embench: biquad_cascade_df2T_f32, config: sos3_n128)
- *DCT*: Discrete cosine transform (Embench: dct4_512_f32, dct4_2048_f32)
- *Block LMS adaptive FIR*: least-mean-square-adaptive filter operating on a block of real data (Embench: lms_f32, config: taps256_n128).

Table 2 shows the full set for DSP 1.0, showing the intensity of branching, compute, and memory per benchmark.

Our initial focus has been on processing floating-point data. Floating-point multiplication and addition functional units are available on several embedded processors; however, it is important to have fixed-point or fractional data types for many application domains. We will be releasing these versions of key algorithms in the future.

Note that our naming convention (dsp_fir_block_f32) means 32-bit floating-point data. We will add naming support of fixed-point algorithms. For example, dsp_fir_block_q31 could be used to represent the processing of 32-bit data with 31 fractional bits.

BENCHMARKING CODE SIZE

While evaluating code size might not seem like rocket science, It is actually more difficult than performance, which only needs a stopwatch.

The obvious code size issue is the compiler flags, which can be set to optimize performance or to shrink code size. You'd like everyone reporting code size to make the same choice on compiler flags (see the "What is the impact on code size of compiling for highest performance, and vice versa?" section).

TABLE 2. The full set for DSP 1.0

Embench DSP 1.0 Name	Origin	C LOC	Branching	Memory	Compute
biquad_cascade_df2T_f32 (sos3_n1)	BDTi	605	15%	36%	46%
biquad_cascade_df2T_f32 (sos3_n128)	New	636	9%	19%	72%
dct4_512_f32	New	1,044	8%	37%	56%
dct4_2048_f32	New	1,426	7%	38%	56%
fir_f32 (taps256_n1)	BDTi	799	15%	31%	54%
fir_f32 (taps256_n128)	BDTi	829	14%	29%	57%
lms_f32 (taps256_n1)	New	630	14%	33%	52%
lms_f32 (taps256_n128)	New	675	13%	33%	53%
rfft512_f32	BDTi	889	3%	37%	61%
rfft2048_f32	BDTi	1,273	2%	37%	60%

Each row represents an Embench DSP benchmark and configuration. This table uses the same numerical boundaries and colors as Embench IOT 2.0 in Table 1. As one might expect, on average the DSP benchmarks have lower branching intensity and higher memory intensity than the IOT benchmarks.

The subtle issue is what to do about libraries; embedded programs are so small that libraries often are the majority of the code. For CoreMark, the main code shrinks from 93% of the total when using newlib-nano to 51% when using glibc. (The embedded processor measured is the ARM Cortex-M4.) As mentioned previously, libraries dominate Dhrystone code. Main code shrinks from 54% of the total for newlib-nano down to only 9% for glibc. Clearly, the choice of the library can overwhelm the size of the embedded benchmark you're trying to measure.

Embench sidesteps this problem by simply requiring subtraction of the library code size from the final published code size results. With regard to compiler flags, we record size results for both optimizing for performance and optimizing for code size.

Our benchmark programs are pure C11 routines that are linked against a main harness and minimal support library. To measure the library overhead of a given toolchain, we compile said harness and library against an empty benchmark. We ensure that the toolchain cannot optimize library code away by exporting every symbol. The resulting executable's size is only attributed to library and runtime overhead, and can be subtracted from each benchmark to yield a representative size measurement.

Alas, as code size is not officially part of the legacy benchmarks, it's up to individuals to decide how to measure it. It is unlikely that one can confidently compare code size for CoreMark or Dhrystone from independent evaluations, as the choices of libraries and compiler flags lead to drastically different numbers.

EMBENCH IOT 2.0 IN ACTION

Given a more precise evaluation tool, there are numerous interesting questions that we can now answer accurately. To keep this introduction concise, we'll only work through a few examples but suggest others for readers to pursue.

How do performance and code size change as one adds optional instruction extensions to RISC-V?

RISC-V is an unusual instruction set since it has a base set (RV32I) and then many optional extensions. Embench shows an improvement of 1.81 if we include the optional multiply and divide instructions, and CoreMark suggests the benefit is 2.58. However, Dhrystone suggests they only add 1%! If you believe Dhrystone, embedded applications hardly benefit from including multiply and divide hardware. Since that hardware can be expensive in tiny cores, Dhrystone encourages omitting such support even though it is vital to many embedded programs.

What is the impact on code size of compiling for highest performance, and vice versa?

For the latest GCC compiler targeting RISC-V, performance drops 16% if optimizing for code size, and code size increases 54% if optimizing for performance.

How do the 32-bit ARM instruction set and the 32-bit RISC-V instruction set compare for code size?

Figure 2 gives the answer for GCC versions 14.1. The geometric mean of the code size ratios suggests ARM is on average ~9% smaller than RISC-V, with a geometric standard deviation range from 0.78 to 1.08, or from 22% smaller to 8% larger. The "Benchmarking Code Size" section and Figure 2 show that the libraries linked with the code have a much bigger effect than any inherent difference in code size between the two architectures and compilers.

Detailed analysis of the differences allows us to see the strengths and weaknesses of the two architectures and their compilers, and hence what can be improved. For example, the ARMv7-M has specialist instructions that benefit cryptographic algorithms, unlike the RV32IMC. (RISC-V does have an extension with specialist cryptographic instructions, but it wasn't included in the microprocessor we measured.)

ARMv7-M has load multiple and store multiple instructions to save and restore registers on a function call. To keep the instruction set and implementations simple, RISC-V does not have instructions that make multiple memory accesses in its base instruction set. To reduce code size, the compiler can invoke an intermediate function to save and restore registers, via the use of the `–msave–mrestore` flag. Figure 2 is based on that option. This optimization reduces Embench code size by 8% over using -Os without `–msave–mrestore`.

More questions

Here are more interesting questions that Embench IOT 2.0 and DSP 1.0 could address but CoreMark and Dhrystone cannot (space limits prevent including answers here):

1. How much do comparisons differ based on the compiler, for

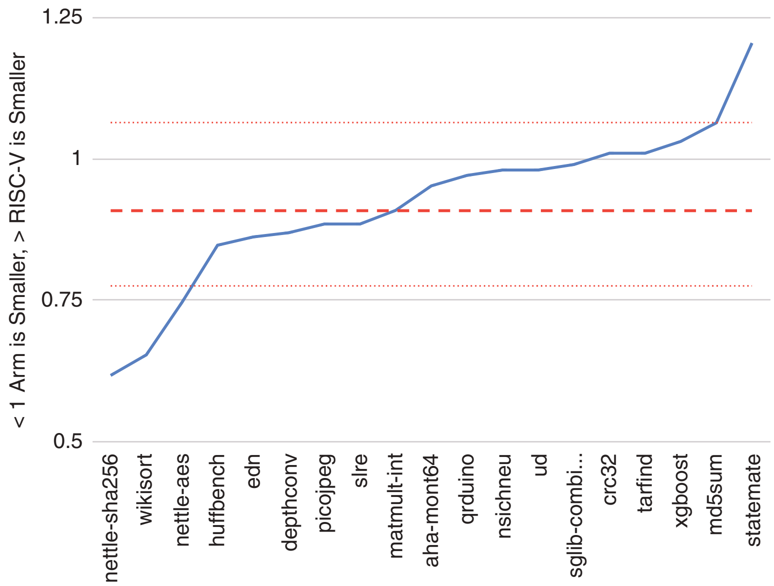


FIGURE 2. Relative code size for GCC version 14.1 for ARMv7-M and RV32IMC for Embench 2.0.

example, GCC versus low level virtual machine (LLVM)?

2. How do other instruction sets fare, for example, Cadence's Tensilica or Intel's x86?
3. ARM actually has several instruction sets. How do the code sizes of ARM v7 and v9 differ?

FOUR USE CASES BEYOND TRADITIONAL BENCHMARKING

The Zephyr Project uses Embench to automatically test embedded system-on-chip (SoC) CPUs.[15] Conveniently, tests are generated automatically on a server as part of a continuous integration software development process. It collects information about the SoC, runs Embench, and then records performance per core and for the whole system.

Similarly, SystemC-based Virtual Prototype embeds Embench to simplify design space exploration of IOT systems based on RISC-V.[16]

A third example is compiler tuning. GCC has >1,000 internal parameters that adapt the compiler to a computer. Since Embench takes only a few seconds to compile, to explore the parameter space using iterative compilation,[17] we compiled it nearly 100,000 times in 24 hours, shrinking code size another 7%.

A final example is WebAssembly, a byte-code virtualization technology finding use cases beyond the browser. Strong separation properties and easy portability make WebAssembly interesting for embedded systems.[18] In this use case, researchers picked representative benchmarks from the Embench suite to learn that one interpreter style favors branch-dominated applications. As single programs, CoreMark and Dhrystone benchmarks cannot offer such focused insights.

RELATED WORK

Dhrystone debuted in 1984 as a synthetic program written in Ada that tried to model the static frequency of programming language constructs—assignment states, call/return, if, …—in systems programming applications.[4] To get Dhrystone MIPS (DMIPS), one divides the Dhrystone score by 1,757 (score of the supposedly 1 MIPS VAX 11/780).

The EDN Embedded Microprocessor Benchmark Consortium (EEMBC) was brought together by EDN magazine in response to the limitations Dhrystone and its poor reproducibility.[7] As a historical note, the dissatisfaction with Dhrystone as a benchmark in the late 1980s also inspired the original SPEC CPU benchmark 35 years ago.

Several companies in embedded processor development were the founding members. The consortium developed a suite of realistic benchmark programs to be used for marketing and development. EEMBC did not specify a single summary score. To ensure the soundness of results, EEMBC initially insisted that published benchmark results must be certified and that members of the consortium could only publish results for their own processors.

Although the benchmarks were extensively used by processor developers and tools vendors in the 2000s, they never became popular in papers or promotions and failed to displace Dhrystone, perhaps due to the proprietary nature of the benchmarks, publication restrictions, and high cost.

In an attempt to readdress its original goal of replacing Dhrystone, in 2009 EEMBC produced CoreMark, an open source program which combined following four algorithms into a single synthetic program:

1. find and sort in list processing
2. matrix manipulation
3. state machine
4. cyclic redundancy check.

These algorithms are similar to four found in the original EEMBC suite. While free and easy to port, it is not presented as a suite, and it has not evolved in 15 years.

CoreMark-Pro is a more recent suite with five integer and four floating-point programs, but a 50x larger memory footprint than Embench (4 MiB). While CoreMark use remains widespread, neither the original EEMBC suite nor CoreMark-Pro became as popular. (The EEMBC organization was recently acquired by the nonprofit organization SPEC to become the SPEC Embedded Group.)

Like the SPEC CPU benchmark, Geekbench is aimed at much larger systems than Embench, with the smallest target being a smartphone.

Academics tried following many of the best benchmark practices to develop a free and open benchmark suite for the embedded market. The most popular was MiBench,[19] which debuted in 2001 but has not been updated. Despite MiBench having twice as many programs as Embench (38 versus 19), 20 years later we concluded that 95% were not relevant. This fact highlights the importance of an organization to evolve a benchmark over time. Embench 1.0 included only two of the 38 MiBench kernels. Embench 2.0 has just one, as MBench's cubic kernel was one of the four dropped from Embench 1.0.

A related effort is the development of real-time benchmarks, which can be important for embedded applications. The most prominent is the Mälardalen Worst-Case Execution Time (WCET) benchmarks[20] from 2010.

BEEBS was an earlier attempt to build on MiBench, WCET, and similar efforts.[11] The MAGEEC project created it to provide a balanced set of real-world programs to support compiler optimization for energy efficiency. Its 80 benchmarks provided the starting point for the Embench 1.0 suite, largely a subset of BEEBS (see Table 1).

Sadly, none of these efforts displaced CoreMark or Dhrystone. As mentioned previously, one lesson learned from past failures is the importance of having an organization that evolves and sustains the benchmark over time. Embench is being developed by a committee of the Free and Open Source Silicon (FOSSi) Foundation (https://fossi-foundation.org/). Like we recently did for Embench IOT 2.0 and Embench DSP 1.0, we intend to revisit the current version and release new versions every few years to keep the benchmark up to date. (If you'd like to help Embench evolve, contact the Embench vice-chair Ray Simar at ray.simar@rice.edu).

Despite the tremendous growth of small, embedded devices, the two most popular embedded benchmarks are dated and severely flawed. Hardware and software engineers are forced to choose between making changes that help real programs or optimizing irrelevant benchmarks that help marketing.

The advantages of Embench IOT 2.0 and DSP 1.0 for evaluating embedded computing over CoreMark and Dhrystone are as follows:

- a suite of programs rather than a single program
- measuring code size (including its careful definition)
- reporting a single performance summary number *plus standard deviation*
- requiring thorough result documentation to ensure reproducibility
- having an organization to evolve and sustain the benchmark over time.

Embench is following in the footsteps of good benchmarks from other fields, just adding a code size metric that is critical for embedded devices.

Embedded computing has been relying on bad benchmarks for decades, but there is hope. Embench IOT 2.0 and DSP 1.0 highlight processor and compiler differences that these synthetic programs either can't detect or exaggerate. If practitioners start quoting Embench scores alongside CoreMark and Dhrystone, then we can start the transition to a place where real progress is measured and rewarded. When we get there, 2009's CoreMark and 1984's Dhrystone can finally be retired to their proper places in benchmark history. C

REFERENCES

1. D. A. Patterson, "For better or worse, benchmarks shape a field," *Commun. ACM*, vol. 55, no. 7, p. 104, 2012.
2. J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*. Amsterdam, The Netherlands: Elsevier, 1990.
3. S. Gal-On and M. Levy, "Exploring CoreMark – A benchmark maximizing simplicity and efficacy," Embedded Microprocessor Benchmark Consortium, El Dorado Hills, CA, USA, 2012. [Online]. Available: https://www.eembc.org/techlit/articles/coremark-whitepaper.pdf
4. R. P. Weicker, "Dhrystone: A synthetic systems programming benchmark," *Commun. ACM*, vol. 27, no. 10, pp. 1013–1030, 1984.
5. Arm, "Arm Cortex-R processor comparison table," 2021. [Online]. Available: https://www.arm.com/-/media/Arm%20Developer%20Community/PDF/Cortex-A%20R%20M%20datasheets/Arm%20Cortex-R%20Comparison%20Table.pdf
6. A. Waterman, private communication, 2024.
7. A. R. Weiss, "Dhrystone benchmark: History, analysis, "scores" and recommendations," white paper, EEMBC Certification Lab., El Dorado Hills, CA, USA, 2002. [Online]. Available: https://www.docjava.com/courses/cr346/data/papers/ECLDhrystoneWhitePaper%20(1).pdf
8. D. A. Patterson, "Embench™: Recruiting for the long overdue and deserved demise of Dhrystone as a benchmark for embedded computing," *Computer Architecture Today Blog*, Jun. 11, 2019. [Online]. Available: https://www.sigarch.org/embench-recruiting-for-the-long-overdue-and-deserved-demise-of-dhrystone-as-a-benchmark-for-embedded-computing/
9. P. J. Fleming and J. J. Wallace, "How not to lie with statistics: The correct way to summarize benchmark results," *Commun. ACM*, vol. 29, no. 3, pp. 218–221, 1986.
10. J. R. Mashey, "War of the benchmark means: Time for a truce," *SIGARCH Comput. Archit. News*, vol. 32, no. 4, pp. 1–14, 2004, doi: 10.1145/1040136.1040137.
11. J. Pallister, S. Hollis, and J. Bennett, "BEEBS: Open benchmarks for energy measurements on embedded platforms," 2013, *arXiv:1308.5174*.
12. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in

ABOUT THE AUTHORS

DAVID PATTERSON is a Distinguished Engineer with Google, Mountain View, CA 94043 USA, and Professor Emeritus at the University of California, Berkeley, Berkeley, CA 94706 USA. His research interests include domain specific architectures, the carbon footprint of computers, and helping shape AI for the public good. Patterson received a Ph.D. in computer science from the University of California, Los Angeles. Contact him at pattrsn@berkeley.edu.

JEREMY BENNETT is a founder of Embecosm, SO15 2AF Southampton, U.K. His research interests include compilers, embedded computing, and open source software and hardware. Bennett received a Ph.D. in computer science from the University of Cambridge. Contact him at jeremy.bennett@embecosm.com

MARY BENNETT is a software tool chain engineer at Embecosm, SO15 2AF Southampton, U.K. Her interests include compilers, software libaries, and open source hardware. Bennett received a masters degree in computer science from the University of Surrey. Contact her at mary.bennett@embecosm.com.

HÉLÈNE CHELIN is a software tool chain engineer at Embecosm, SO15 2AF Southampton, U.K. Her interests include compilers, cybersecurity, and networks. Chelin received a bachelors degree in cybersecurity, network and system from the EPITA School of Engineering in Paris. Contact her at helene.chelin@embecosm.com.

DAVID HARRIS is the Harvey S. Mudd Professor of Engineering Design at Harvey Mudd College, Claremont, CA 91786 USA. His research interests include microprocessor design, computer arithmetic, and undergraduate education. Harris received a Ph.D. in electrical engineering from Stanford University. Contact him at harris@g.hmc.edu.

JENNIFER HELLAR is an engineer with Cirrus Logic, Austin, TX 78701 USA. Her research interests include digital design, computer architecture, signal processing, and embedded system design. Hellar recieved a master's degree in electrical engineering from Rice University. Contact her at jlhellar@proton.me.

WILLIAM JONES is an AI engineering lead with Embecosm, SO15 2AF Southampton, U.K. His research interests include AI and Bayes explainability and uncertainty. Jones received a Ph.D. in computer science from the University of Kent. Contact him at william.jones@embecosm.com.

KONRAD MORON is a Ph.D. student at Technische Universität München, 81377 Munich, Germany. His research interests include formal methods, decision procedures/model checking, and programming languages. Moron received a B.Sc. in computer science from Hochschule München University of Applied Sciences. Contact him at k.moron@mailbox.org.

PAOLO SAVINI is a software tool chain engineer at Embecosm, SO15 2AF Southampton, U.K. His research interests include compilers, side channel security and linux device drivers. Savini received a masters degree in computer engineering from the University of Pavia, Italy. Contact him at paolo.savini@embecosm.com.

ROGER SHEPHERD is a founder of Chipless, BS6 6YW Bristol, U.K. His research interests include high-performance processors, embedded processors, and parallel processors. Shepherd received a degree in mathematics from Warwick University. Contact him at rog@rcjd.net.

RAY SIMAR is a professor in the practice in the Electrical and Computer Engineering Department, Rice University, Houston, TX 77251 USA. His research interests include digital signal processing, machine learning, computer architecture, and human metabolism. Simar received a MS in digital signal processing from Rice University. He is a Fellow of IEEE and an ACM Distinguished Engineer. Contact him at ray.simar@rice.edu.

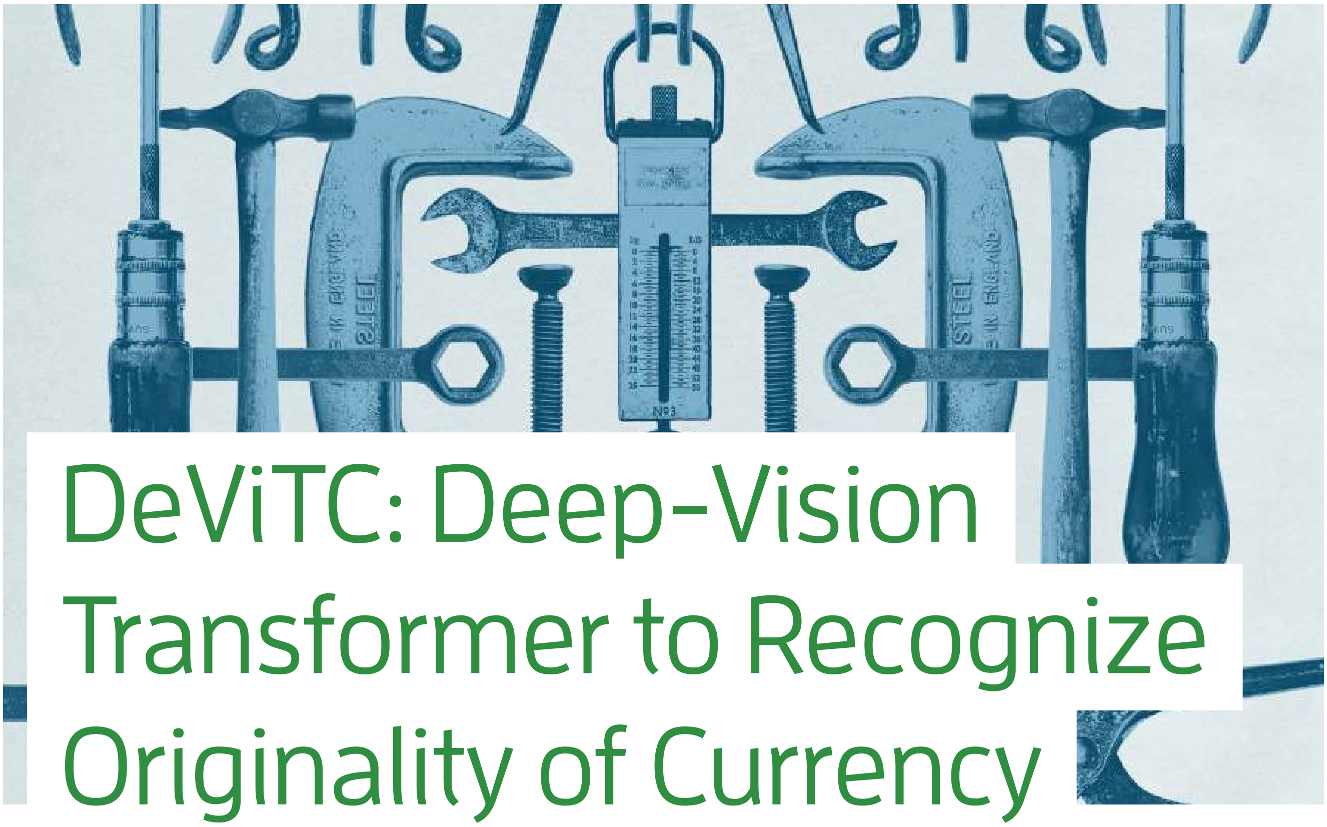
ZACHARY SUSSKIND received a Ph.D. in electrical and computer engineering from The University of Texas at Austin, Austin, TX 78712 USA. His research interests include arithmetic-free machine learning, computer architecture, and extreme edge computing. He is a Member of IEEE. Contact him at zsusskind@utexas.edu.

STEFAN WALLENTOWITZ is a professor with Hochschule München University of Applied Sciences, 80335 Munich, Germany. His research interests include computer architecture, edge computing, embedded system security, embedded runtime systems. Wallentowitz received a Ph.D. in electrical engineering from Technical University Munich. Contact him at stefan.wallentowitz@hm.edu.

Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 785–794.

13. R. Simar. *New DSP Extensions to the Embench Benchmark Suite*. Dec. 29, 2022. [Online Video]. Available: https://www.youtube.com/watch?v=5Xjkn7XaYIA
14. J. Eyre and J. Bier, "DSP processors hit the mainstream," *Computer*, vol. 31, no. 8, pp. 51–59, Aug. 1998, doi: 10.1109/2.707617.
15. "Automatic CPU testing with Embench," *Zephyr Project*, Nov. 5, 2020. [Online]. Available: https://www.zephyrproject.org/automatic-cpu-testing-with-embench/
16. V. Herdt, D. Große, and R. Drechsler, "Fast and accurate performance evaluation for RISC-V using virtual prototypes," in *Proc. Des., Autom. Test Europe Conf. Exhib.*, 2020, pp. 618–621, doi: 10.23919/DATE48585.2020.9116522.
17. Z. Pan and R. Eigenmann, "Fast and effective orchestration of compiler optimizations for automatic performance tuning," in *Proc. Int. Symp. Code Gener. Optim.*, 2006, pp. 12–332, doi: 10.1109/CGO.2006.38.
18. S. Wallentowitz, B. Kersting, and D. Dumitriu, "Potential of WebAssembly for embedded systems," in *Proc. 11th Mediterranean Conf. Embedded Comput.*, 2022, pp. 1–4, doi: 10.1109/MECO55406.2022.9797106.
19. M. R. Guthaus et al., "MiBench: A free, commercially representative embedded benchmark suite," in *Proc. 4th Annu. IEEE Int. Workshop Workload Characterization. WWC-4 (Cat. No.01EX538)*, 2001, pp. 3–14, doi: 10.1109/WWC.2001.990739.
20. J. Gustafsson et al., "The Mälardalen WCET benchmarks: Past, present and future," in *Proc. 10th Int. Workshop Worst-Case Execution Time Anal.*, 2010, pp. 136–146, doi: 10.4230/OASIcs.WCET.2010.136.

Digital Object Identifier 10.1109/MC.2025.3553275



DeViTC: Deep-Vision Transformer to Recognize Originality of Currency

Dulari B. Gajjar, Prisha Faldu, Deep Rameshbhai Kothadiya, Aayushi Pushpakant Chaudhari, and Nikita M. Bhatt, Charotar University of Science and Technology (CHARUSAT)

This paper presents a deep learning-based banknote recognition system using the Vision Transformer (ViT) architecture. Evaluated on Indian currency and four datasets, the ViT's attention mechanism enhances accuracy with data augmentation. Comparisons with ResNet, VGG, GoogleNet, and EfficientNet demonstrate its robustness, aiding financial inclusion.

Technological advancements in computer vision and machine learning have enhanced currency recognition systems' accuracy. This article presents a deep learning approach for banknote recognition using the Vision Transformer (ViT) architecture. The ViT's attention mechanism captures long-range dependencies and hierarchical features from currency images. The study evaluates the ViT model on Indian currency denominations and four other datasets, utilizing data augmentation techniques like image rotation and distortion to improve robustness. Testing different patch sizes and attention heads, the ViT model achieves high precision on the Indian currency dataset. The approach is also compared with models like ResNet, VGG, GoogleNet, and EfficientNet. The article further discusses the potential applications and challenges of currency recognition programs in aiding the visually impaired and promoting financial inclusion.

INTRODUCTION TO CURRENCY RECOGNITION

With the rapid advancement in technologies like computer vision and machine learning, the detection of various currencies is becoming increasingly sophisticated and accurate. In recent years, digital forms of financial transactions have gained in popularity due to their enhanced

Digital Object Identifier 10.1109/MC.2024.3514151
Date of current version: 28 April 2025

security features, such as encryption and biometric authentication, which can be more secure than traditional cash transactions. These digital systems also provide significant benefits for visually impaired individuals, offering greater accessibility through features like screen readers and voice commands. However, despite these advancements, cash transactions remain prevalent in many regions, and their recognition and security are critical challenges.

The algorithms discussed in this article primarily focus on currency recognition, which, while valuable, may be limited in scope compared to digital solutions. Furthermore, implementing these algorithms often requires expensive hardware and software, which may not be accessible to low-income populations, presenting a barrier to widespread adoption. Another concern is the potential for tampering with currency recognition systems, which could allow counterfeit money to be incorrectly identified as legitimate. These limitations should be considered when evaluating the applicability of such algorithms.

Given these considerations, the article will explore both the potential of currency recognition algorithms and the challenges associated with their implementation while recognizing that digital transactions are increasingly seen as a more secure and accessible alternative. Deep learning techniques, along with computer vision, help these systems to analyze visual data, extract meaningful features,[1] and make accurate decisions to classify different denominations of coins and banknotes. It also becomes a critically important research topic to facilitate visually impaired individuals in managing their financial transactions.

Security measures are a major factor in ensuring the authenticity of detected banknotes and coins in currency detection. Counterfeit currency proves to be a major threat to such currency-detecting algorithms. If counterfeit currency is introduced, then it could lead to a substantial economic crisis.[2] Therefore, prioritizing security becomes a crucial task while developing such systems. This could be done by advanced algorithms, rigorous validation processes, high-resolution images, and using in-grain security features like watermarks or holograms in currencies.[3] Thus, this aspect of security not only guarantees safeguarding the interests of all the parties involved but also highlights the significant role played by the aspect of security in global economic systems.

Currency recognition technology plays a paramount role in different industries, with a significant focus on aiding visually impaired individuals. Apart from the verification of banknotes and coins, it is widely used in banking operations, retail sales, and self-service kiosks, solving the problem of human errors. It is also actively involved in foreign trade and finance, providing currency conversion and compliance with the rules and regulations.[4] Nevertheless, its biggest contribution can be seen in facilitating the visually impaired to take full control of their finances by themselves, thus promoting their social inclusivity and financial independence.

Distinguishing currency recognition by employing cutting-edge computer vision and deep learning significantly improves the accuracy of currency detection and classification.[5] This technology is used not only for its technical abilities but also to confront the threats of counterfeiting and maintaining the integrity of financial transactions as well as its ability to empower visually impaired people, allowing them to take control of their finances. As we explore the depth of currency recognition, we not only unveil its technical attributes but also its social ramifications, indicating the coming of a world that is easier to use and inclusive.

Detecting Indian currency using computer vision faces several research challenges, including the variety in currency note designs, frequent updates, and the need to distinguish notes from complex backgrounds and cluttered environments.[6] Challenges also arise from dealing with low-quality images, noise, and artifacts as well as the need to detect subtle differences in counterfeit notes. Limited datasets and class imbalance can hinder the effective training of models, while real-time processing demands speed and efficiency, especially on resource-constrained devices. Ensuring generalization across diverse conditions, adapting to design changes, and integrating with other technologies are also significant hurdles. To address these issues, advanced algorithms, data augmentation, transfer learning, and collaboration with financial institutions are essential for creating robust and adaptable detection systems.

This article examines the capabilities and limitations of the currency detection system through rigorous analysis, aiming to contribute to the ongoing research. The "Dataset" section introduces the dataset acquisition method and various preprocessing techniques applied to it using computer vision. In the "Methodology" section, advanced deep learning algorithms and ViT[7] are discussed technically. The simulation, results, and analysis of other datasets are presented under the "Results" and "Comparative Study" sections. Finally, the last section concludes the study.

LITERATURE REVIEW

Singh et al.[8] proposed a currency recognition system for Indian currency. They applied segmentation, which helped in noise removal from the images. Then, they applied a method called visual bag of words and scale-invariant feature transform (SIFT) descriptor, thus achieving an accuracy of 96.7%. Wang et al.[9] proposed a methodology consisting of five stages: image acquisition, preprocessing, feature extraction, classification, and result analysis. Using such methods, they achieved an accuracy rate of 93.4.

Yadav et al.[10] proposed a Currency Recognition System Based on Oriented FAST and rotated the YoloV3 algorithm to the Indian currency dataset of six different denominations. The Sobel algorithm was used to extract the inner and outer edges of the image after the preprocessing step. YoloV3 was then used for clustering, thus achieving a notable accuracy of 91.2%. Caytuiro-Silva et al.[11] addressed the challenge of the real-time detection of multinational banknotes with a dataset of 9,315 images of Peruvian banknotes. It consisted of denominations of 10, 20, 50, and 100 Peruvian soles and applied machine learning and deep learning models. Using that, they enhanced the accuracy of banknote processing systems for currency detection and identification in real time.

Swain et al.[12] proposed a novel deep learning framework for automated Brazilian coin classification. It included a Repetitive Feature Extractor Convolution Neural Network (RFE-CNN), which leverages state-of-the-art convolutional neural networks. This model achieved a notable accuracy of 98.34% through multistage processing and transfer learning.

Foysal et al.[13] utilized deep learning and computer vision techniques using modified versions of the YOLOv7 and YOLOvS algorithms for multiclass object detection and currency classification, respectively. They aimed to support blind and visually impaired individuals with their intelligent systems, which achieved a remarkable mean average precision of 94.6% in YOLOvS for currency classification and 91.4% in YOLOv7 for object detection.

Jawale et al.[14] proposed a system to aid visually impaired individuals to avoid fraud by currency recognition. They took the input via camera and applied deep learning algorithms like VGG16, ResNet50, AlexNet, Detectron2, YOLOv7, and natural language processing for text extraction. With VGG16 achieving the highest accuracy of 97%, YOLOv7 was also precise in detecting different currencies.

METHODOLOGY

ViT

The proposed study leverages a deep-ViT network for effective paper currency recognition, even on small datasets. Deep-ViT manages long-range dependencies and relies on an attention mechanism, enabling it to capture global input–output relationships without recurrence. The network operates in three stages: linear encoding, multihead attention, and multilayer perceptron (MLP). Deep-ViT extracts hierarchical features through these stages, tokenizing image patches and refining details for high-accuracy recognition. This structure enables the model to effectively classify paper currency with limited data, enhancing accuracy and robustness in currency recognition. Figure 1 illustrates the proposed architecture that uses the hybrid Transformer model for recognizing the currency.

The Transformer follows the encoder-decoder architecture, with the ability to process sequential data in parallel without relying on any recurrent network. Figure 2 represents the detailed architecture of multihead attention used in the proposed model. The success of Transformer models has largely benefited from the self-attention mechanism, which is proposed to capture long-range relationships between the sequence's elements. ViT utilizes the encoder module of the Transformer to perform classification by mapping a sequence of image patches to the semantic label. Unlike the conventional CNN architectures that typically use filters with a local receptive field, the attention mechanism employed by the ViT allows it to attend to different regions of the image and integrate information across the entire image.

Let $N = \{X_i, Y_i\}_{i=1}^{c}$, where c is a possible class for currency, X represents the currency image, and y represents the corresponding class of that currency. The proposed architecture is composed of an embedding layer and an encoder with multihead self-attention (MSA) followed by an MLP feed-forward neural network (FNN). Initially, an image X from the training set (for simplicity, we omit the image index i) is subdivided into nonoverlapping patches. Each patch is viewed by the Transformer as an individual token. Input image X as $(h * w * c)$ having width w, height h, and number of channels c and then extract patches, each of dimension $c * p^2$. This forms a sequence of patches $(x_1, x_2, \ldots, x_n)$ of length n, with $n = h * w/p^2$. Typically, the patch size p is chosen as 16 × 16 or 32 × 32, where a smaller patch size results in a longer sequence and vice versa.[15]

Linear embedding

Before feeding the sequence of patches into the encoder, each patch x_i is linearly projected into a lower-dimensional feature space using a trainable linear projection matrix W and a bias vector b with $Z_i = \mathrm{ReLU}(x_i W + b)$ Here, Z_i is the projected feature vector for the patch x_i, and ReLU is the rectified linear unit activation function.[16] It is linearly projected into a vector of the model dimension d using a learned embedding matrix E. The embedded representations are then concatenated together along with a learnable classification token v_{class} that is required to perform the classification task. The embedded image patches are viewed by the Transformer as a set of patches without any notion of their order. To keep the spatial arrangement of the patches as in the original image, the positional information E_{pos} is encoded and appended to the patch representations. The resulting embedded sequence of patches with the token Z_0 is given in the following equation:

$$Z_o = [V_{\text{class}}\colon x_1\ E,\ X_2 E,\ \ldots, x_n E] + E_{\text{pos}} \in R^{(n+1)d},\ E \in R^{p^2 cd}. \qquad (1)$$

Encoder

Each encoder layer independently processes the input sequence to capture different levels of abstraction. The self-attention sublayer computes attention scores between each pair of linear embedding in the input sequence to capture dependencies and relationships between them. The FFN sublayer applies a fully connected feed-forward neural network to each token independently, enabling nonlinear transformations and feature extraction. This sublayer applies a fully connected feed-forward neural network to each token independently, enabling nonlinear transformations and feature extraction.

The proposed network uses an MLP for a fully connected feed-forward

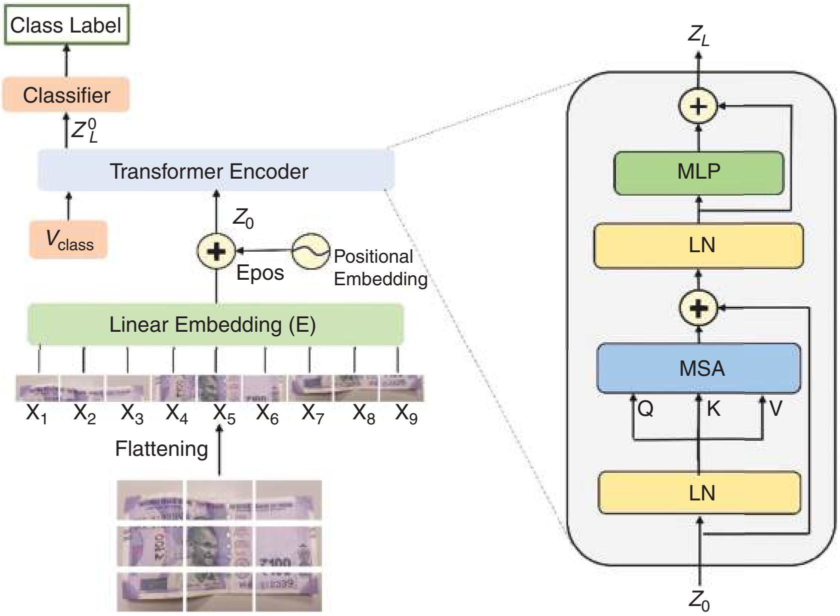


FIGURE 1. The proposed architecture of the hybrid Transformer model for currency recognition. LN: layer norm; MSA: multihead self-attention.

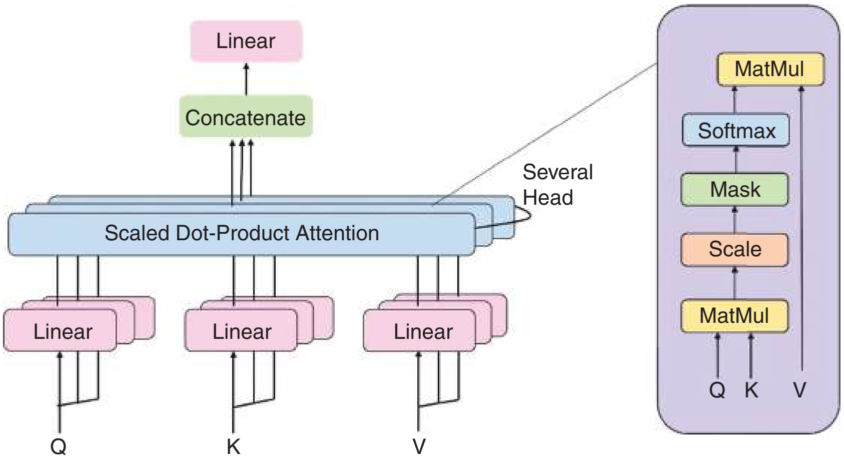


FIGURE 2. The conceptual architecture of multihead attention used in the proposed study.

dense block. The output of the linear embedding layer resulting in sequence patches $X = [x_1, x_2, \ldots, x_n]$ is transmitted to the Transformer encoder. The proposed deep Transformer consists of two subblocks of the self-attention layer and MLP. Sequential execution of the encoder block can be formulated as

$$Z_l^{TM} = MSA(LN(z_l - 1)) + z_{l-1} \quad (2)$$

$$Z_l = MSA(LN(Z_l^{TM})) + Z_l^{TM} \quad (3)$$

where L is the number of normalization layers. At the final layer, we take each element Z_L^0 and forward it to the final MLP classification as formulated in

$$y = LN(Z_L^o) \quad (4)$$

where $l = 1 \ldots L$.

The self-attention mechanism is mainly divided into three categories: compute matrix, compute attention, and concatenate attention score. We linearly project the input sequence X into three separate matrices. Query (Q), key (K), and value (V) matrices using learnable weight matrices, W_q, W_k, and W_v, are formulated in the following[17]:

$$Q = X * W_q \quad (5)$$

$$K = X * W_k \quad (6)$$

$$V = X * W_v. \quad (7)$$

The dimensions of K, Q, and V are shown as Nd_k where d_k stands for the dimensionality of the key, query, and value vectors. Next, we compute the attention scores A between each pair of tokens using the dot product between the query and key matrices, scaled by the square root of the dimensionality of the key vectors $\sqrt{d_k}$ calculated as the following equation to mitigate the impact of large values[18]:

$$A = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right). \quad (8)$$

Attention score A is used to compute a weighted sum of the value matrix V, which represents the attended representation for each token as attention $(A) = A * V$. Finally, we concatenate the attended representations from all tokens and pass them through a linear projection layer to obtain the output of the self-attention mechanism formulated as the following equation. The output of the self-attention mechanism represents the contextualized representations of the input image patches, which can then be aggregated (for example, by taking the mean or using a classification head) to make predictions about the image class.

$$MSA(z) = \mathrm{Concat}\,(SA_1(z), SA_2(z), \ldots, SA_n(z))W,$$
$$W \in \mathbb{R}^{hD_kD}. \quad (9)$$

We modified the proposed architecture into three variants: altering patch sizes, learning layers, and self-attention heads. The base variant uses 16 × 16 patches with eight attention heads, while the large variant features 32 × 32 patches and 12 heads.

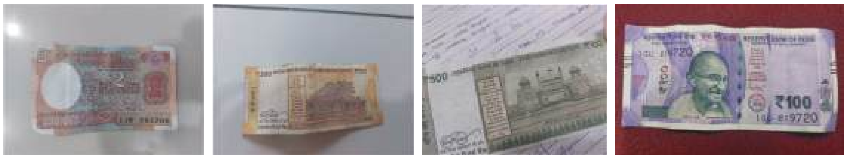

(a) (b) (c) (d)

FIGURE 3. (a)–(d) Original images of various currency denominations.

DATASET

Our dataset includes more than 9,000 smartphone-captured images of Indian currency notes, covering nine denominations [rupees (Rs) 1, 2, 5, 10, 20, 50, 100, 200, and 500]. For Rs 10, 50, and 100, both old and new designs (Reserve Bank of India) are included. Images were taken under varied lighting (sunlight, dim, and room lit) and backgrounds (plain and with text/numbers), simulating real-life scenarios with both new and worn notes. Each denomination has more than 1,000 images, enhancing algorithm learning for currency recognition. Figure 3 provides a glimpse of the dataset.

Simulation of the proposed methodology used different augmentation techniques, such as image rotation and distortion, to enlarge and diversify the samples. Specifically, image rotation was simulated using angles of 10°, 20°, 30°, and 40° and also contained different moderate factors ranging from 0.5 to 2.0 to introduce variability in the dataset. We manually sorted the images into labeled folders for easy handling of data while training and testing. Also, to standardize the input data for training, all images were resized to a uniform size of 224 × 224 pixels with three color channels, which created consistency across the dataset and allowed for efficient deep learning algorithm processing.

Other datasets

In addition to our custom Indian currency dataset, we analyzed four more datasets: Ethiopian birr, Ghana cedi, Peruvian sol, and Thai baht. The Ethiopian dataset included denominations of five, 10, 50, 100, and 200 birrs; the Ghanaian dataset had 1–200 cedi notes; the Peruvian dataset included 10, 20, 50, and 100 soles; and the Thai dataset featured 20–1,000 baht bills. Using the same augmentation techniques as for Indian currency, we applied rotation and distortion to these datasets for comprehensive analysis. Table 1 summarizes the datasets for currency recognition.

RESULTS

The simulation of the proposed methodology uses a system with a Core i7, GeForce RTX 3080, and 32 GB of RAM. TensorFlow and the Keras library of Python were used to implement the model. Simulation of the proposed study uses 20 epochs with 16 batch sizes having variations in the self-attention head. The last layer of the network uses softmax, and the internal layers use ReLU as an activation function.

The ViT model was performed with learning epochs set at 20 and 30 and batch size fixed at 32. A 70-30 train-validation split, where 70% of the dataset is employed for training and the remaining 30% for validating, has been chosen. For testing purposes, we used unseen data for a better understanding of the model's performance. Table 2 represents the accuracies of different models with various epochs.

During the implementation of the ViT, we also tried out different numbers of attention heads (two, four, six, eight, and 12) and patch sizes (8 × 8, 16 × 16, 32 × 32, and 24 × 24). The amount of heads determines the model's attention to different parts of the input image, while the patch size affects the level of detail of the feature extraction. Therefore, increasing attention heads improved accuracy by capturing finer data dependencies, and larger patch sizes also led to higher accuracy by encompassing more information and better feature representation and recognition. Table 3 represents the accuracy of the proposed architecture for different datasets. Figure 4 represents the resultant currency denominations in the form of a confusion matrix.

TABLE 1. The dataset summary.

Dataset name	Number of classes	Average images per class without augmentation	Average images per class with augmentation	Average resolution
Indian	9	540	4,355	224 × 224
Ethiopian	5	400	3,300	256 × 256
Ghana	8	280	2,100	2,080 × 1,560
Peruvian	8	850	5,500	640 × 480
Thai	5	230	1,850	1,024 × 1,024

TABLE 2. The accuracy of different models with different numbers of epochs.

Model variant	Accuracy with 20 epochs	Accuracy with 30 epochs
ViT	99.13%	99.56%
GoogleNet	98.33%	98.58%
ResNet50	98.69%	98.92%
ResNet30	99.38%	99.61%
VGG	92.56%	93.15%
EfficientNet	91.99%	92.08%

TABLE 3. The accuracy of the proposed architecture over different datasets.

Currency/model	ViT	ResNet50	ResNet30	GoogleNet	VGG
Ethiopian	94.72	98.63	98.24	97.49	83.06
Peruvian	56.62	53.12	72.41	74.26	34.79
Thai	90.78	89.78	91.11	79.21	66.39
Ghana	99.5	95	99.4	99.39	95

COMPARATIVE STUDY

In this study, we conducted a comparative analysis of our custom dataset using various deep learning models, including ResNet30,[19] ResNet50,[20] VGG16,[21] GoogleNet,[22] and EfficientNet.[23] Each of these models possesses unique architectural characteristics, contributing to their efficacy in image classification tasks. Figure 5 represents the accuracies and losses from training and validation, and Figure 6 represents a comparative analysis of the proposed model with reference to various heads and number of patches.

Through rigorous experimentation, we aim to identify the most suitable model for our dataset and gain insights into its strengths, weaknesses, and applicability in real-world currency recognition scenarios. In implementing all these architectures with custom adjustments on our dataset, a standardized approach was adopted. Each model was trained for 20 epochs with a batch size of 32, using a learning rate of 0.001, a weighted decay at 0.1, and a dropout rate of 0.2.

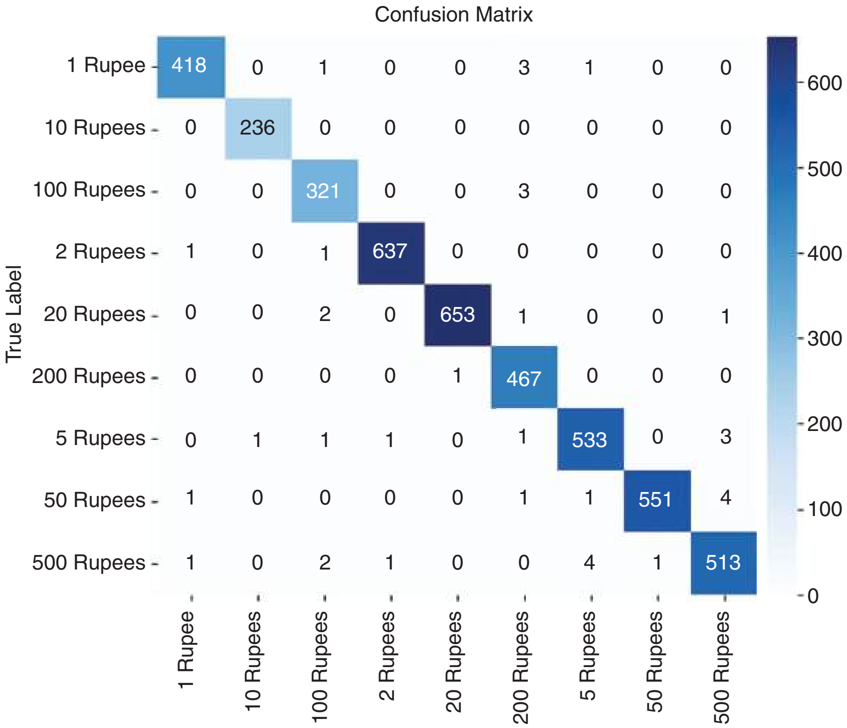


FIGURE 4. The recognized results of currency denominations in the form of a confusion matrix.

This article provides a comprehensive analysis of the ViT architecture for paper money identification, leveraging its self-attention mechanism to effectively capture long-range dependencies and extract hierarchical features from currency images. The ViT model achieved a remarkable

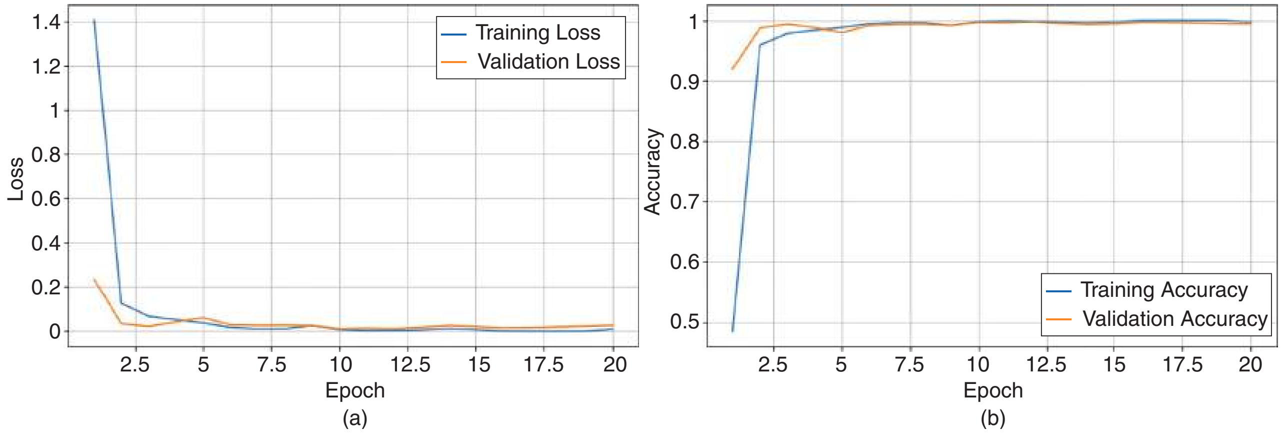


FIGURE 5. The resultant losses and accuracies. (a) A graphical representation of training and validation loss. (b) A graphical representation of training and validation accuracy. (a) Training and validation loss. (b) Training and validation accuracy.

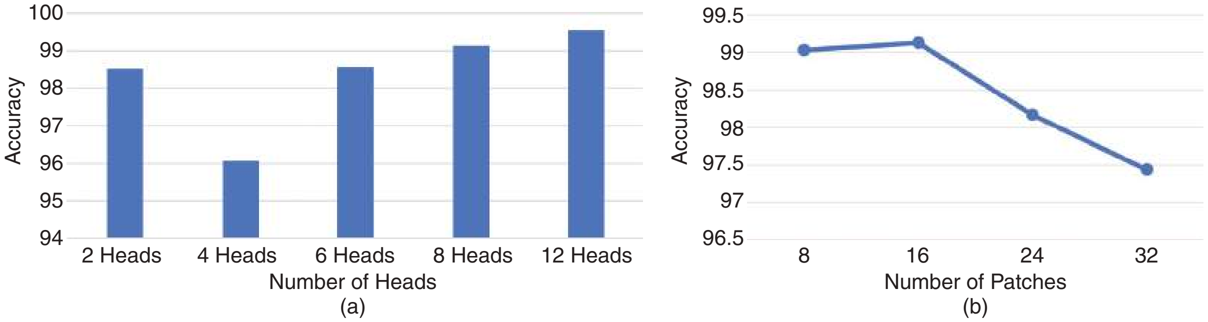


FIGURE 6. A comparative analysis of the proposed model. (a) A comparative analysis with a different number of heads. (b) A comparative analysis with a different number of patches.

99.56% accuracy on a specially created Indian currency dataset, outperforming other leading deep learning algorithms like ResNet, VGG, GoogleNet, and EfficientNet, and demonstrating its superior ability to present intricate details and patterns in currency design. The research also explored the impact of various modes, such as patch size and attention units, on model precision, and it examined the ViT model's performance on datasets from countries like Ethiopia, Ghana, Peru, and Thailand. While the results varied across different datasets, the ViT model showed potential for generalization and scalability. Overall, this study highlights the ViT model's capability to achieve high precision in currency recognition, addressing counterfeiting challenges and promoting financial inclusivity for visually impaired individuals while also paving the way for more advanced research in currency recognition. C

REFERENCES

1. J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Mach. Learn. Appl.*, vol. 6, Aug. 2021, Art. no. 100134, doi: 10.1016/j.mlwa.2021.100134.
2. Y. Hamid, S. Elyassami, Y. Gulzar, V. R. Balasaraswathi, T. Habuza, and S. Wani, "An improvised CNN model for fake image detection," *Int. J. Inf. Technol.*, vol. 15, no. 1, Nov. 2022, doi: 10.1007/s41870-022-01130-5.
3. A. Mukundan, Y.-M. Tsao, W.-M. Cheng, F.-C. Lin, and H.-C. Wang, "Automatic counterfeit currency detection using a novel snapshot hyperspectral imaging algorithm," *Sensors*, vol. 23, no. 4, pp. 2026–2026, Feb. 2023, doi: 10.3390/s23042026.
4. S. Sarkar and A. K. Pal, "A new approach to fuzzy logic analysis of Indian currency recognition," *J. Print Media Technol. Res.*, vol. 11, no. 2, pp. 119–128, 2022.
5. S. M. Bahrani, "Deep learning approach for Indian currency classification," *Int. J. Eng. Appl. Sci. Technol.*, vol. 5, no. 6, pp. 335–340, 2020.
6. K. S. S. Reddy, G. Ramesh, C. Raghavendra, C. Sravani, M. Kaur, and R. Soujanya, "An automated system for Indian currency classification and detection using CNN," in *Proc. E3S Web Conf.*, vol. 430, 2023, p. 9, doi: 10.1051/e3sconf/202343001077.
7. A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021, *arXiv:2010.11929*.
8. S. Singh, S. Choudhury, K. Vishal, and C. V. Jawahar, "Currency recognition on mobile phones," in *Proc. 22nd Int. Conf. Pattern Recognit.*, 2014, pp. 2661–2666, doi: 10.1109/icpr.2014.460.
9. H. H. Wang, Y. C. Wang, B. L. Wee, and M. Chen, "Novel feature extraction and representation for currency classification," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 33, no. 1, pp. 275–284, Oct. 2023, doi: 10.37934/araset.33.1.275284.
10. S. Yadav, Z. A. Ansari, and K. G. Singh, "Currency detection for visually impaired," *J. Emerg. Technol. Innov. Res.*, vol. 7, no. 5, pp. 999–1002, 2014.
11. N. E. Caytuiro-Silva, J. M. Peña-Alejandro, E. G. Castro-Gutierrez, J. Sulla-Torres, and B. Maraza-Quispe, "Annotated Peruvian banknote dataset for currency recognition and classification," *Data Brief*, vol. 51, Dec. 2023, Art. no. 109715, doi: 10.1016/j.dib.2023.109715.
12. D. Swain et al., "A deep learning framework for the classification of Brazilian coins," *IEEE Access*, vol. 11, pp. 109,448–109,461, 2023, doi: 10.1109/access.2023.3321428.
13. N. Uddin Foysal, T. J. Thamid, M. S. Uddin, M. F. Islam, M. N. Islam, and F. A. Faisal, "Advancing AI-based assistive systems for visually impaired people: Multi-class object detection and currency classification," *in Proc. Int. Conf. Inf. Commun. Technol. Sustain. Develop. (ICICT4SD)*, 2023, pp. 438–442, doi: 10.1109/icict4sd59951.2023.10303628.

ABOUT THE AUTHORS

DULARI B. GAJJAR is a student with the U & P U. Patel Department of Computer Engineering, Chandubhai S Patel Institute of Technology, Charotar University of Science and Technology (CHARUSAT), Changa 388421, India. Her research interests include machine learning, deep learning, and computer vision. Contact them at 21ce031@charusat.edu.in.

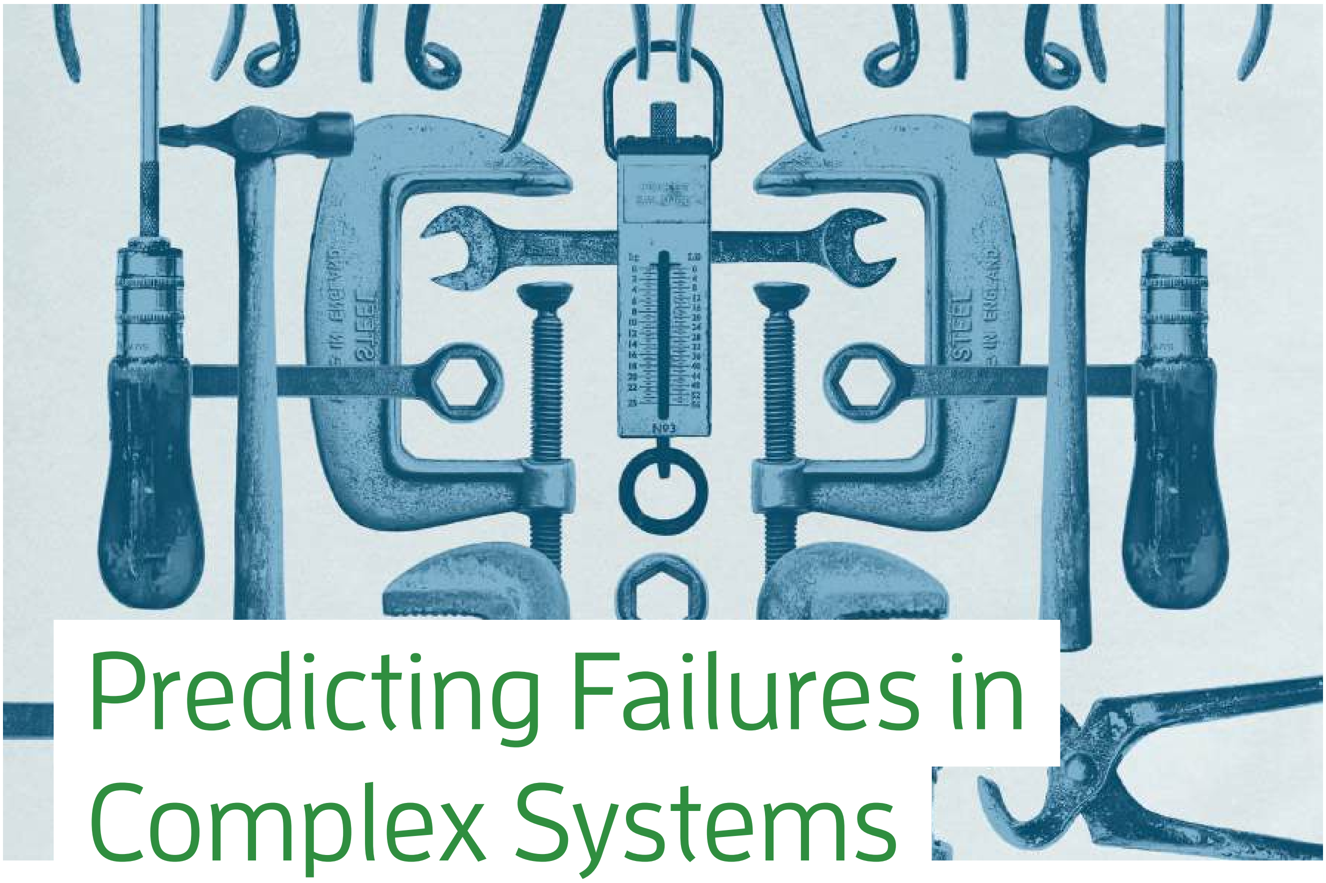
PRISHA FALDU is a student with the U & P U. Patel Department of Computer Engineering, Chandubhai S Patel Institute of Technology, Charotar University of Science and Technology (CHARUSAT), Changa 388421, India. Her research interests include natural language processing and computer vision. Contact them at 21ce028@charusat.edu.in.

DEEP RAMESHBHAI KOTHADIYA is an assistant professor at U & P U. Patel Department of Computer Engineering, Chandubhai S Patel Institute of Technology, Charotar University of Science and Technology (CHARUSAT), Changa 388421, India, and he is also a researcher at Prince Sultan University Riyadh 12435, Saudi Arabia. His research interests include machine learning, deep learning, and computer vision. Kothadiya earned a Ph.D. in computer engineering from CHARUSAT. He is also a technical reviewer of many Web of Science- and Scopus-indexed journals and conferences Contact him at deepkothadiya.ce@charusat.ac.in.

AAYUSHI PUSHPAKANT CHAUDHARI is an assistant professor in the U & P U. Patel Department of Computer Engineering, Chandubhai S Patel Institute of Technology, Charotar University of Science and Technology (CHARUSAT), Changa 388421, India. His research interests include machine learning, deep learning, and computer vision. Chaudhari earned a Ph.D. from CHARUSAT. She also serves as a technical reviewer for several prestigious journals and conferences, reflecting her active involvement and expertise in the academic and research community. Contact her at aayushichaudhari.ce@charusat.ac.in.

NIKITA M. BHATT is the head of the computer engineering department at the U & P U. Patel Department of Computer Engineering, Chandubhai S Patel Institute of Technology, Charotar University of Science and Technology (CHARUSAT), Changa 388421, India. Her research interests include machine learning, deep learning, and natural language processing. Contact them at nikitabhatt.ce@charusat.ac.in.

14. A. Jawale, K. Patil, A. Patil, S. Sagar, and A. Momale, “Deep learning based money detection system for visually impaired person,” in *Proc. 7th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, 2023, vol. 5, pp. 173–180, doi: 10.1109/iciccs56967.2023.10142586.
15. D. Kothadiya et al., “Enhancing fingerprint liveness detection accuracy using deep learning: A comprehensive study and novel approach,” *J. Imaging*, vol. 9, no. 8, Aug. 2023, Art. no. 158, doi: 10.3390/jimaging9080158.
16. A. Jariwala, A. Chaudhari, C. Bhatt, and D.-N. Le, “Data quality for AI tool: Exploratory data analysis on IBM API,” *Int. J. Intell. Syst. Appl.*, vol. 14, no. 1, pp. 42–56, Feb. 2022, doi: 10.5815/ijisa.2022.01.04.
17. D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A.-B. Gil-González, and J. M. Corchado, “Deepsign: Sign language detection and recognition using deep learning,” *Electronics*, vol. 11, no. 11, Jun. 2022, Art. no. 1780, doi: 10.3390/electronics11111780.
18. D. R. Kothadiya, C. Bhatt, A. Chaudhari, and N. Sinojiya, “GujFormer: A vision transformer-based architecture for Gujarati handwritten character recognition,” in *Proc. Int. Conf. Adv. Data-Driven Comput. Intell. Syst.*, 2024, pp. 89–101, doi: 10.1007/978-981-99-9524-0_8.
19. B. Koonce, *Convolutional Neural Networks with Swift for Tensorflow*. Berkeley, CA, USA: Apress, 2021.
20. X. Tian and C. Chen, “Modulation pattern recognition based on Resnet50 neural network,” in *Proc. IEEE 2nd Int. Conf. Inf. Commun. Signal Process. (ICICSP)*, 2019, pp. 34–38.
21. K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *arXiv:1409.1556*.
22. C. Szegedy et al., “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1–9.
23. M. Tan and Q. V. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. 36th Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 6105–6114.



Predicting Failures in Complex Systems

João R. Campos and **Ernesto Costa**, University of Coimbra
Marco Vieira, University of North Carolina at Charlotte

Online failure prediction is a technique to predict incoming failures in the near future. This allows for preemptive measures to avoid, or at least mitigate, their consequences. This article shows how recent advances make it now possible to develop accurate failure predictors for complex systems.

Software is currently used to execute critical and sensitive tasks on a daily basis. Due to the growing complexity and the need for continuous development, it is not possible to detect every fault before deployment. These residual faults can lead to failures at runtime, posing significant risks. The Meta outage in 2024 cost approximately US$100 million,[1] and software faults have been identified as major contributors to various incidents, such as the 2020 AWS Kinesis outage, the 2021 Facebook outage, and various recent aerospace disasters,[2] as well as Boeing 737 Max crashes.[3] Over the years, several techniques have been proposed to support the development of dependable software.[4] The main problem is that many of them do not scale properly to the complexity dimension of modern software, while others have limited coverage or are too expensive.

Online failure prediction (OFP) is a fault tolerance technique that uses the current state of the system to predict incoming failures at runtime. Such predictions can then be acted upon to avoid or mitigate their consequences. This can avoid unnecessary risks and costs, while simultaneously improving availability and reliability attributes, but despite the great potential, it is yet to become widespread.

Digital Object Identifier 10.1109/MC.2025.3526342
Date of current version: 28 April 2025

Several challenges prevented it from being applied to complex systems in the past (such as a lack of failure data or the infeasibility of identifying relevant performance metrics). However, recent advances,[5] such as developments in artificial intelligence (AI) and machine learning (ML) algorithms, combined with tailored validation techniques and comprehensive approaches, now make it possible to create accurate and detailed failure predictors for modern complex systems, such as full-fledged operating systems (OSs). This can even allow for differentiated preemptive mechanisms that depend on the type of incoming failure.

These breakthroughs have the potential to mark a change in the development of the systems of the future. In fact, they allow for widespread adoption of OFP by enabling development teams to systematically create failure prediction mechanisms specifically designed for their systems, without incurring the considerable costs typically associated with other fault tolerance techniques.

DEPENDABLE COMPUTING

Dependability is a composite concept that comprises different attributes, such as availability, reliability, safety, integrity, and maintainability. Throughout the years, several techniques have been designed to support the development of more dependable systems. They can be mainly divided into two large groups: 1) fault avoidance, aims for fault-free systems and includes techniques to prevent or remove faults, and 2) fault acceptance, which accepts that faults are inevitable and deals with their existence, encompassing techniques for fault forecasting and fault tolerance.[4] When a *fault* is activated, it can cause the system to deviate from its correct state, which is known as an *error*. An error can propagate and alter the service provided by the system, which is defined as a *failure*.

One of the main challenges that are currently faced is that due to the growing complexity and size of software, fault avoidance techniques such as code reviews and software testing do not scale well or in an affordable manner. In an attempt to overcome this issue, various works have tried to use AI and ML to predict the occurrence of faults (for example, Alsina et al.[6]) or to detect faults for removal (for example, Neysiani and Babamir[7]). Despite these advances, fault avoidance is becoming increasingly more difficult.

On the other hand, fault acceptance techniques, mainly fault tolerance, have been successfully used in a variety of critical domains. Fault tolerance accepts the fact that faults will make their way to production and thus focuses on avoiding or mitigating their effects. Some of the techniques that have been particularly successful focus on fault masking, which aims to hide the occurrence of faults while continuing to provide the correct service. Redundancy and diversity, at various levels from hardware to code, have achieved remarkable results. However, these techniques also have their limitations. For example, although quite effective, redundancy can be costly due to the need to replicate parts of the system, and it is only as effective as its coverage.

OFP

OFP is a fault tolerance technique that tries to predict at runtime whether a failure, potentially caused by some residual fault that escaped the testing phase, will occur in the near future. To achieve this, it can leverage various sources of data, such as logs and runtime system metrics. It is particularly relevant in the context of large and complex systems, where the risks and costs associated with a failure are not negligible, in the presence of residual faults that cannot be tolerated by existing fault-tolerance mechanisms. By using OFP to predict incoming failures, it is possible to take preemptive measures to avoid, or at least mitigate, the effects of such failures.[8] Even if it is not possible to completely avoid failure, such information can be used to initiate recovery mechanisms, expediting system recovery. These actions have the potential to reduce losses due to system outages, thus improving availability and reliability and directly influencing the trustworthiness of the system.

Over the years, multiple approaches have been proposed for OFP. Various alternative sources of data can be used to develop failure predictors,[8] especially when considering large complex systems. Due to its ease of use, some works have relied on system logs for failure prediction and reliability engineering, as thoroughly surveyed in He et al.[9] However, log analysis either relies on the premise that the logs of the target system are detailed, structured, and consistent or assumes the that system owner will develop and implement the necessary logging mechanisms. To achieve a more automated, generic, and detailed characterization of the system, one of the most prominent approaches is to continuously monitor system metrics (for example, CPU and memory). The premise is that in addition to causing a failure, an error may cause the system to behave erratically (also known as a *symptom*).

In practice, OFP relies on different kinds of information, such as past failure data, to train the predictor, and the current state of the system, to make predictions.[8] An illustration of the task can be seen in Figure 1. A prediction made at time t targets a window starting at time $t + \Delta t_l$ and lasting for Δt_p (Δt_l and Δt_p are usually referred to as the *lead time* and *prediction window*). At time *t*, the model should predict whether a failure is going to occur during the prediction window. A prediction is correct if the failure event occurs at least once within the prediction interval. The width of the Δt_l defines how far ahead the failure is to be predicted and represents the ideal lead time to avoid or initiate repair mechanisms.

CHALLENGES AND LIMITATIONS

There are a series of challenges that have prevented the widespread use of OFP. One of the main issues is that failures are rare, and as a result, failure data are typically not available. Collecting data from real systems would take years and a considerable number of unhandled failures and would by then likely be deprecated. Moreover, failure events may still not exist or be unknown for new systems, and as such, it is impossible to develop failure predictors during the software development lifecycle. As a result, most work on OFP focused on smaller components for which data are easily accessible or where it is possible to clearly define a small set of quality metrics, such as disk (for example, Zhang et al.[10]) or job failures in cloud applications (for example, Jassas and Mahmoud[11]).

Even when data are available, creating accurate failure predictors is not a trivial task. Existing techniques such as reliability growth and analytical methods have limitations,[12] considering that modern systems are characterized by hundreds or thousands of system metrics. Moreover, complex systems also present a variety of different and complex failure modes, which might require differentiated treatment.

ML has allowed for steadily overcoming some of the previous issues. However, these are interdisciplinary subjects that require expert knowledge. This presents some challenges as it is necessary to evaluate multiple algorithms, but the choice of which techniques to use and how to properly assess alternative solutions depend on several aspects. Furthermore, to avoid biased or unrepresentative performance estimates, it is necessary to follow well-defined and established procedures, using techniques that take into consideration the specific characteristics of OFP. ML models have also been shown to be sensitive to context drift and small variations in the data, which can be particularly challenging as the behavior and usage of production systems evolve over time.

OFP for complex systems

The combination of the various challenges meant that system-level OFP was unreachable, up until recently. Recent advances[5] have provided the knowledge and tools to create accurate predictive models for complex systems, such as a full-fledged OSs. These techniques can be used in systems for which failure data are not yet available and for which there are so many system metrics and failure modes that it is not possible to a priori specify a set of diagnostic indicators.

Developing predictive models for complex systems involves three main steps: 1) generating realistic failure data leveraging fault injection techniques, 2) training predictive models taking into consideration the specific characteristics of the problem, and 3) fairly assessing and comparing the performance of the various failure predictors considering the operational requirements of the target system.

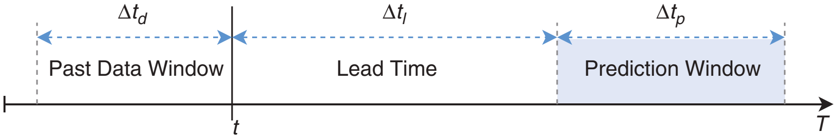


FIGURE 1. The time relations in OFP, adapted from Salfner et al.[8]

Generating failure data

The first step is to generate data to support the development of the predictive models. Fault injection has been accepted and shown to be a viable alternative to generate realistic failure data for complex systems in feasible time.[13] In practice, it consists of injecting realistic faults (which can be at both the hardware and the software level) and then monitoring the system behavior throughout the experimental process. When some of these faults are activated, they can lead to errors, which can subsequently lead to failures.

A component that is essential in this process is the workload. The workload represents the tasks that the system will be executing when generating the data. Because the workload influences the behavior of the system (that is, a CPU-intensive workload will have a different execution profile than an I/O-intensive), it needs to be similar to the workload that is expected to occur in production.

As the predictive models will be trained using the generated data, it is important that both the fault model and the selected workload are representative of the system where the predictors will operate. To ensure the representativeness of the data, this process should follow a well-defined methodology and systematic approach.[13,14] Recent generative algorithms have also shown promising results and have the potential to support this process in the near future, although the feasibility of such an approach has not yet been proven.

Developing predictive models

The next step focuses on the development of the predictive models for the target system. Recent contributions have put forward detailed methodologies and novel validation methods to support the development of ML-based failure predictors.[15]

Different ML algorithms will model the data differently. This is the premise of the no free lunch theorem, which states that on average, there is no algorithm that is better for all instances of all classes of problems. This way, different algorithms and techniques should be considered and experimented, to find the one that can best model the underlying problem. The task of OFP has two main parameters that need to be defined, Δt_l and Δt_p (the lead time and the prediction window, respectively). They define the distance that exists between the prediction and the event. Recent works have shown that different failure modes will start exhibiting failures at different moments in time and thus have different optimal Δt_l and Δt_p.

One of the most important steps of this process is estimating the performance of the model. This is paramount, as it will influence the expectation of usefulness of the predictors. If not done properly, it can lead to a biased selection process. Due to the specific event-based time series nature of the OFP problem and its data, traditional validation techniques do not apply. Models developed using these techniques failed to perform in deployment, hindering the perception of usefulness of OFP. To obtain a more realistic performance estimate and drive the selection process toward models that are able to generalize properly, the experiment-wise leave-one-out cross-validation (ELOOCV)[15] validation approach for OFP can be used. It is also necessary to manage the performance of the models under small perturbations in the data, leveraging adversarial and robustness optimization techniques, which will inevitably occur in production systems.

Benchmarking predictive models

Aligned with the process of developing predictive models, we need to compare alternative solutions. To ensure a fair and sound comparison, one should follow a set of well-defined guidelines and processes.

One of the main decisions is the selection of which performance metric to use for the comparison. This choice needs to take into account the characteristics of the data (for example, class imbalance), but it should also be made based on the technical needs and impact of the system in an organization. This can be seen by means of requirements in terms of the level of dependability that should be satisfied and the cost of mitigating the predicted failures before their occurrence. For a home banking system, it is better to select the predictor with a higher detection rate, even if it raises more false alarms than others (within some acceptable bounds), since unpredicted failures lead to significant losses. On the other hand, for a less critical corporate site, we might want a model that does not raise too many false alarms since the cost of dealing with them may be high compared with the risk posed by the failures.

Within the benchmarking process it is also important to consider that in a practical scenario, no preemptive mechanisms will be triggered based on an individual alert. Instead, it will look at a sequence of closely time-related predictions to establish confidence, from the perspective of a system administrator.

Establishing a proper experimental process that allows for a fair assessment and comparison according to benchmarking principles is not trivial. Simplified approaches lead to incomplete or irreproducible results, often biased by inadequate choice of metrics or techniques. It is necessary to adhere to a comprehensive methodology for benchmarking OFP solutions[16] that provides detailed guidelines and considerations.

Following an iterative cycle, the performance of the models should be continuously monitored after deployment for performance degradation due to changes in the underlying process. If necessary, new data should be generated or included in the dataset, and the models should be updated.

PREDICTING FAILURES IN LINUX

To demonstrate the high potential for improving the dependability of modern systems, we present an overview of how to develop failure predictors for the Linux OS, without a priori knowledge or predefined indicators. Linux is often the chosen platform in the most varied applications, from small embedded to large enterprise-grade systems. The state of a system running Linux is characterized by hundreds of system metrics, from networking to I/O and memory management. Given its complex inner workings, it also exhibits a diverse set of failure modes. To support this analysis, we consider three different workloads that are representative of different usage scenarios.

Failure data

Fault injection is used to generate failure data, following the methodology detailed in Campos and Costa.[13] We generate the data through a comprehensive fault injection campaign with a realistic fault model that is applicable to the Linux OS. Three representative workloads (*io*, *cpu*, and *matrix*) are selected for this study, taken from Ubuntu.[17] Naturally, when developing failure predictors for a given system, the workload needs to be representative of what is expected to occur in production. The system is monitored for various failure modes, such as *crash* (OS crashes), *hang* (OS hangs), *cpu/execution-related* (for example, *invalid opcode*), *memory-related* (for example, *segmentation fault*), and *kernel-related* (for example, *recursive fault*) failures.

To collect data, we use the Netdata tool, which collects several system metrics every second, providing a detailed characterization of the system at any given time. Afterward, we remove constant and transient features, resulting in a dataset with 371 features. Using this process, a total of 4,487, 4,432, and 4,498 experiments are executed for the *cpu*, *matrix*, and *io* workloads, respectively, in a relatively short amount of time. It is worth noting, however, that not all injected faults lead to failures that are useful for OFP (for example, a system that crashes immediately after activating the fault is impossible to predict). This approach allows for creating comprehensive and rich data that are both specific and representative of realist data for the target system, contrary to most existing works, which leverage open datasets or simply focus on benign data.

Predicting failures

When creating predictive models, it is important to consider a comprehensive set of ML and techniques, from “classic” and well-proven to state-of-the-art algorithms. For this study, several algorithms (for example, SVM, MLP, and XGBoost) are selected and fine-tuned through a comprehensive grid search. Various preprocessing techniques, from dimensionality reduction (for example, PCA and highly correlated features), to reduce the complexity, to sampling techniques (for example, SMOTE and ADASYN), to handle the imbalance in the data, are also evaluated. Due to the fact that most features in the dataset are based on different scales, a Z-score standardization was applied to the data. Several values were considered for the lead time and prediction window parameters.

In a classification problem, the samples that are correctly predicted are known as true positive (TP) and true negative (TN). The positive samples (that is, failures) that are predicted as negatives (that is, nonfailures) are false negative (FN), and the opposite is false positive (FP). For this analysis, we consider the requirement of a critical system, where the goal is to detect as many failures as possible while still taking into account the number of false alerts (that is, a model that is constantly raising false alerts is not acceptable). To achieve this, the assessment and ranking of the various models are done considering the *informedness* performance metric. This is a widely used metric in dependability research, which measures how consistently a predictor predicts the outcomes of both the failures and the nonfailures[18] (informedness = TP/(TP + FN) + TN/(FP + TN) − 1. ELOOCV was used to ensure a proper error estimation, where each experiment is considered for evaluation, closely representing a real scenario.

Due to space constraints, we focus on the failure predictors for the *cpu* workload, although similar results were obtained for the other two workloads. After training, fine-tuning, and benchmarking alternative solutions for the various failure modes, the best predictive models are identified, which can be seen in Figure 2.

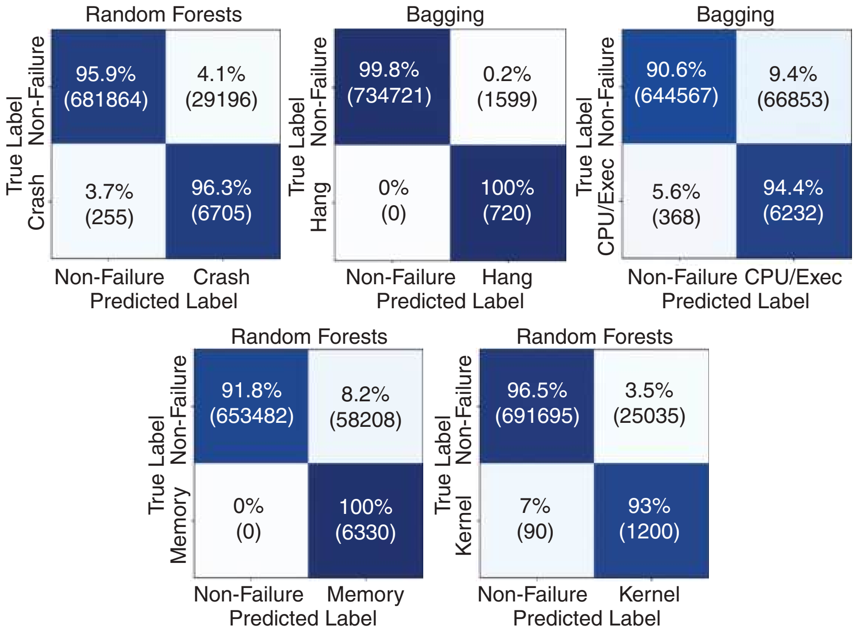


FIGURE 2. Performance by failure mode for the *cpu* workload.

The confusion matrices provide a clear understanding of the performance of the models in terms of correct and incorrect predictions. As can be observed, it is possible to create accurate failure predictors for each failure mode. As an example, it is possible to correctly predict every sample of *hang* failure with a minimal rate of false positives (that is, nonfailure samples predicted as failure), while the *kernel* failure mode has the lowest performance, predicting 93% of failure samples. Something that is important to look for is false positives. Given the fact that most of the time the system will be operating under normal conditions, even 2% or 3% can be too much. After analyzing the false positives, especially for the *cpu/exec* failure mode as it displayed 9.4% false positives, we observe that almost all of them are on samples from other failing runs (that is, other failures being predicted as a different failure mode).

From a practical perspective, no system will take action based on a single alert. The system will only trigger preemptive measures after a given number of alerts within a predefined window. Ultimately, the goal is to assess how effective the models will be in predicting the various failures, from the perspective of the system or system administrator. For this analysis, we set a conservative requirement of having five consecutive alerts (that is, five alerts in the last five seconds). By analyzing the results, we observe that all the failures except one can be predicted by their respective failure predictor. There are, however, some failure predictors that also raise alerts in the presence of other failures. While not ideal, it is likely that some failure modes might exhibit similar symptoms. More importantly, there are no false alerts in any of the healthy runs. Ultimately, out of 190 tests on failure experiments in this study, 185 are correctly predicted, and there is not a single false alert on the 175 tests on healthy runs. This process differs significantly from existing works due to the systematic analysis, evaluation, and comparison of alternative solutions taking into consideration several new aspects, such as the usage scenario and respective adequate metrics, as well as analyzing the task from an administrator's perspective.

Generalizability of OFP solutions

Another important aspect that also deters people from using OFP is how well the predictors will be able to generalize to the variations inherent to production environments.

To assess this, we conduct an adversarial analysis,[16] which aims to identify minimal perturbations to the data that can cause the model to misclassify. Using this approach, we observe that most models are quite robust by default, where the minimal perturbation required for a change is already considerable. It is, however, possible to identify one model that is more sensitive to small variations in the data. In such situations, it is possible to use modified versions of the algorithms or other defensive techniques. To demonstrate this, we use the approach proposed in Chen et al.[19] (a novel robust algorithm based on XGBoost). The number of samples per distortion/model can be seen in Figure 3, where the *x*-axis represents the necessary perturbation (up to 0.3, defined as an acceptable variation in various domains and validated in this case study; for example, for the *average cpu percentage of system apps* feature, whose values range from 0 to 88%, it would mean a maximum change of 0.81%) and the *y* the number of samples that can be perturbed to cause a misclassification. As can be observed, for the original (nonrobust) model it is possible to generate many more adversarial samples within the maximum perturbation and with considerably low perturbations. Conversely, there are significantly fewer adversarial samples for the robust model, which also require higher perturbations, leading to more stable behavior in production.

Because this is such a relevant obstacle (that is, how does a shift in the underlying distribution of the problem affect the performance of the failure

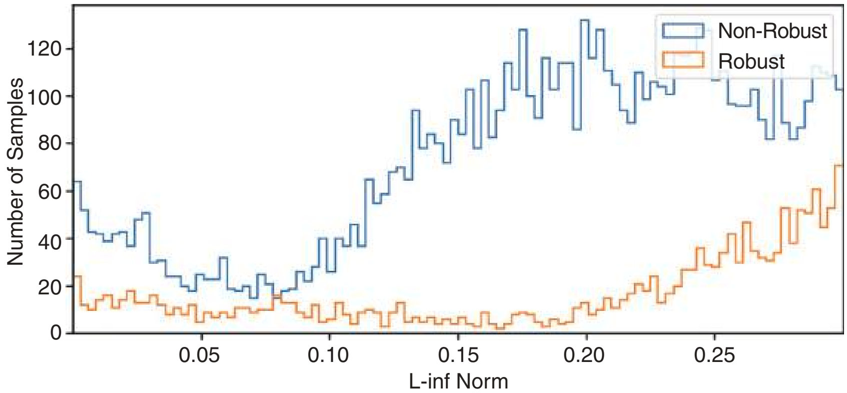


FIGURE 3. The number of samples per L_∞ perturbation (up to 0.3).

predictors?), we go one step farther and measure the predictive performance of the models trained using the *cpu* workload on the experiments from the other two workloads (that is, *io* and *matrix*). Impressively, no false alerts are raised on healthy runs. Additionally, most of the predictive models are able to generalize considerably well, all correctly predicting more than 78% of the failure samples for every failure mode.

These analyses provide a clear interpretation and validate the potential benefits of OFP. Compared with previous works, not only is it now possible to improve the dependability of modern systems by creating accurate failure predictors, but they are also robust to variations in the workload, in terms of both avoiding false alerts and detecting failures. Conversely, models developed using standard validation and training approaches failed to generalize properly, also demonstrating high sensitivity to variations in the data, significantly impacting overall performance.

LESSONS LEARNED AND FUTURE DIRECTIONS

Modern complex systems are characterized by hundreds of system metrics and exhibit intricate behaviors and multiple failure modes. Despite their inherent complexity, it is now possible to develop accurate failure predictors without having a priori knowledge or predefined performance indicators. Leveraging recent contributions, it is possible to overcome the challenges that previously prevented the use of OFP solutions.

As discussed before, following well-defined processes and methodologies, we were able to create accurate predictive models for the Linux OS. Furthermore, although not shown in this article due to space constraints, these advances have also been used to develop OFP solutions for the Windows OS, achieving similar results.[16] Their performance in the presence of workload variations and runtime analysis highlights the potential of these models to work in production environments. OFP is a technique with the potential for an impactful change in software development, enabling teams to seamlessly create and deploy failure prediction mechanisms in their systems within the development lifecycle, without incurring significant costs.

This article presents a holistic perspective on OFP as a valuable fault tolerance tool, applicable across a wide range of critical use cases. Detailed technical implementation insights can be found in Campos et al.[16] and the cited contributions. Although each context will pose specific challenges, the technical resources for developing the models, such as virtualization and ML libraries, are now widely available, and the resulting predictors are self-contained, minimizing integration difficulties and conflicts. Several monitoring tools are also available, which can be easily integrated even in existing systems. Inference requirements for OFP are also unlikely to be lower than 1 s, which is now typically achievable by most ML algorithms.

To face the evolving landscape of modern systems toward larger interconnected, and often distributed, components, researchers should focus on expanding existing contributions to the novel realities. To address the current gap toward unstructured data, multimodal failure predictors might prove necessary to leverage all available data. Recent legislation initiatives have stringent requirements concerning the use of AI in safety-critical domains. As a result, explainable AI techniques will also likely prove necessary to overcome the current explainability gap in complex models.

ACKNOWLEDGMENT

This work was partially supported by Fundação para a Ciência e a Tecnologia (FCT), I.P., in the framework of the Projects UIDB/00326/2020 and UIDP/00326/2020.

REFERENCES

1. S. Liberatore, “Zuckerberg’s million dollar outage: Finance expert reveals Meta lost roughly $100 MILLION during two-hour glitch that took down Facebook, Instagram and Messenger,” *Daily Mail*, Mar. 5, 2024, Accessed: Sep. 25, 2024. [Online]. Available: https://www.dailymail.co.uk/sciencetech/article-13161047/meta-lost-millions-facebook-instagram-outage.html
2. L. E. Prokop, “Historical aerospace software errors categorized to influence fault tolerance,” in *Proc. IEEE Aerosp. Conf.*, 2024, pp. 1–12, doi: 10.1109/AERO58975.2024.10521061.
3. M. McFall-Johnsen, “Catastrophic software errors doomed Boeing’s airplanes and nearly destroyed its NASA spaceship. Experts blame the leadership’s ‘lack of engineering culture’,” *Business Insider*, Feb 29, 2020, Accessed: Sep. 25, 2024. [Online]. Available: https://www.businessinsider.com/boeing-software-errors-jeopardized-starliner-spaceship-737-max-planes-2020-2
4. L. C. Algirdas Avižienis, L. Jean-Claude, and R. Brian, “Basic concepts and taxonomy of dependable and secure

ABOUT THE AUTHORS

JOÃO R. CAMPOS is an assistant professor at the Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal. His research interests include in-depth use of AI and ML for design and development of AI-based solutions for dependability, security, and safety. Campos received a Ph.D. in informatics engineering from the University of Coimbra. He is a Member of IEEE. Contact him at jrcampos@dei.uc.pt.

ERNESTO COSTA is a full professor at the Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal. His research interests include bioinspired AI, developing novel algorithms, and promoting the cross-fertilization of evolutionary computation and ML. Costa received a Ph.D. in computing science from the University Pierre et Marie Curie and a Ph.D. in electronic engineering from the University of Coimbra. Contact him at ernesto@dei.uc.pt.

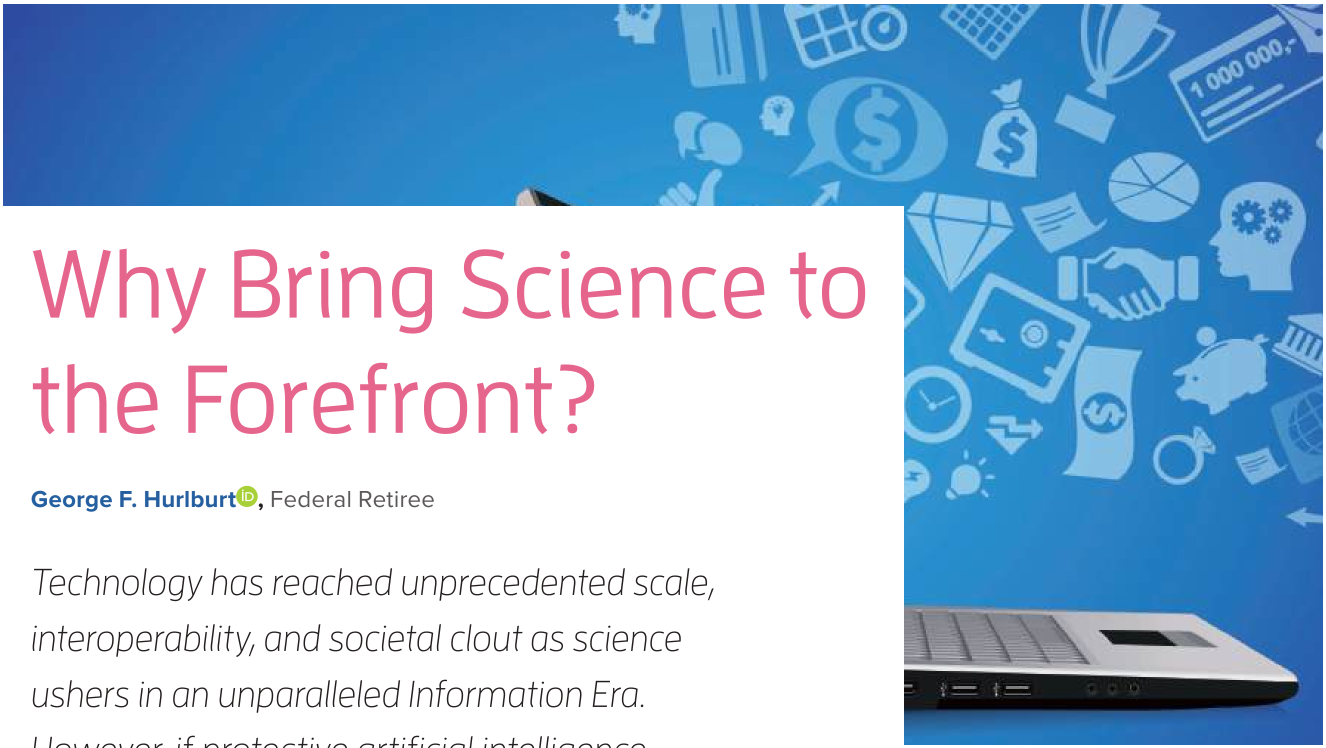
MARCO VIEIRA is a professor at the College of Computing and Informatics, University of North Carolina at Charlotte, Charlotte, NC 28223 USA. His research interests include dependability and security assessment, fault injection, and software testing. Vieira received a Ph.D. in informatics engineering from the University of Coimbra. He serves as the chair of the IFIP WG 10.4 on Dependable Computing and Fault Tolerance. He is a Member of IEEE. Contact him at marco.vieira@charlotte.edu.

computing," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 1, pp., pp. 11–33, 2004, doi: 10.1109/TDSC.2004.2.

5. J. R. Campos, "Advanced online failure prediction through machine learning," Ph.D. dissertation, Universidade de Coimbra, Coimbra, Portugal, 2021.
6. E. F. Alsina, M. Chica, K. Trawiński, and A. Regattieri, "On the use of machine learning methods to predict component reliability from data-driven industrial case studies," *Int. J. Adv. Manuf. Technol.*, vol. 94, nos. 5–8, pp. 2419–2433, Feb. 2018, doi: 10.1007/s00170-017-1039-x.
7. B. S. Neysiani, and S. M. Babamir, "Automatic duplicate bug report detection using information retrieval -based versus machine learning-based approaches," in *Proc. 6th Int. Conf. Web Res. (ICWR)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 288–293, doi: 10.1109/ICWR49608.2020.9122288.
8. F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods," *ACM Comput. Surv.*, vol. 42, no. 3, pp. 1–42, 2010, doi: 10.1145/1670679.1670680.
9. S. He, P. He, Z. Chen, T. Yang, Y. Su, and M. R. Lyu, "A survey on automated log analysis for reliability engineering," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–37, 2021, doi: 10.1145/3460345.
10. J. Zhang et al., "Minority disk failure prediction based on transfer learning in large data centers of heterogeneous disk systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 9, pp. 2155–2169, Sep. 2020, doi: 10.1109/TPDS.2020.2985346.
11. M. S. Jassas, and Q. H. Mahmoud, "Evaluation of a failure prediction model for large scale cloud applications," in *Proc. Can. Conf. Artif. Intell.*, 2020, pp. 321–327.
12. W. Wang, J. Loman, and P. Vassiliou, "Reliability importance of components in a complex system," in *Proc. Annu. Symp. Rel. Maintainability (RAMS)*, Piscataway, NJ, USA: IEEE Press, 2004, pp. 6–11, doi: 10.1109/RAMS.2004.1285415.
13. J. R. Campos and E. Costa, "Fault injection to generate failure data for failure prediction: A case study," in *Proc. IEEE 31st Int. Symp. Softw. Rel. Eng. (ISSRE)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 115–126, doi: 10.1109/ISSRE5003.2020.00020.
14. J. R. Campos, E. Costa, and M. Vieira, "On configuring a testbed for dependability experiments: Guidelines and fault injection case study," in *Proc. Int. Conf. Comput. Saf., Rel. Secur.*, 2020, pp. 419–433.
15. J. R. Campos, E. Costa, and M. Vieira, "Online failure prediction through fault injection and machine learning: Methodology and case study," in *Proc. IEEE 34th Int. Symp. Softw. Rel. Eng. (ISSRE)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 451–461, doi: 10.1109/ISSRE59848.2023.00021.
16. J. R. Campos, E. Costa, and M. Vieira, "Online failure prediction for complex systems: Methodology and case studies," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 4, pp. 3520–3534, Jul./-Aug. 2023, doi: 10.1109/TDSC.2022.3192671.
17. "stress-ng." Ubuntu. Accessed: Oct. 15, 2020. [Online]. Available: https://manpages.ubuntu.com/manpages/artful/man1/stress-ng.1.html
18. D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," 2020, *arXiv:2010.16061*.
19. H. Chen, H. Zhang, D. Boning, and C.-J. Hsieh, "Robust decision trees against adversarial examples," 2019, *arXiv:1902.10660*.

EDUCATION

EDITORS
GEORGE HURLBURT U.S. Federal Service (Retired), USA; gfhurlburt@gmail.com
SOREL REISMAN California State University, USA; sreisman@computer.org



Why Bring Science to the Forefront?

George F. Hurlburt, Federal Retiree

Technology has reached unprecedented scale, interoperability, and societal clout as science ushers in an unparalleled Information Era. However, if protective artificial intelligence guardrails are to be effective, public awareness of modern science and its consequences becomes paramount.

Famed and visionary 20th century scientist Carl Sagan, writing in *The Demon-Haunted World*,[1] prophetically envisioned a future where scientific enlightenment does not exist. In 1995, he wrote:

> "We've arranged a global civilization in which most crucial elements profoundly depend on science and technology. We have also arranged things so that almost no one understands science and technology. This is a prescription for disaster. We might get away with it for a while, but sooner or later this combustible mixture of ignorance and power is going to blow up in our faces."

Despite this dire prophecy, Sagan was both an optimist and a realist. In 1980, he hosted an exquisitely produced, visually stunning, televised science series called *Cosmos*, based on his book of the same name.[2] In it, he intelligently celebrated existing science, always speaking visually to the ordinary citizen.

Today, science has marched on, but the populace has remained largely unaware of the significance of the dramatic, game-changing discoveries over the ensuing decades. New developments in science reveal new dynamics, as described by formerly exotic mathematics. More importantly, this science differentiates the industrial age from the fast-developing information age. Indeed, science has spawned a technological wonderland where, to the uninitiated, everything appears to happen as if by magic. To some, there is concern that, as Sagan predicted, we are becoming digital serfs.[3]

This article examines the key scientific developments fueling the new information age and enabling the technology we depend on. It then explores two popular books dealing with the transition to the information age to illustrate how scientific reality is circumvented. Finally, the article explores educational alternatives to heighten citizen awareness of the discipline-crossing sciences that now underlie indispensable technology.

Digital Object Identifier 10.1109/MC.2025.3544117
Date of current version: 28 April 2025

SCIENCE MARCHES ON

In the early 20th century, Alan Turing set the stage for cryptology, algorithms, computation, and artificial intelligence (AI). His technical interests crossed many disciplines. Two years after ENIAC, the first digital computer, Claude Shannon developed information theory. His theory formalized the role of entropy in quantifying, storing, and transmitting information.[4] This led to many enduring computational practices. Now accepted as the "Father of the Information Age," Shannon demonstrated unification among the stand-alone disciplines of mathematics, statistics, computer science, neurobiology, physics, and electrical and electronic engineering.[5] Moving forward from Shannon's pioneering effort, Figure 1 traces selected scientific movements that shape modern technology.

During the 1950s, Shannon's work helped spawn new fields. Led by Norbert Weiner, cybernetics matured to establish the role of feedback in servo-mechanisms, whether living or inanimate.[6] Australian biologist Karl Ludwig von Bertalanffy established systems science through his general systems theory. Systems science further reinforces the notion of interdisciplinary dependencies, still surviving as a means by which one can unify various fields that have converged since the early days of computing.[7]

> "We might get away with it for a while, but sooner or later this combustible mixture of ignorance and power is going to blow up in our faces."

Complexity

These new fields paved the way for complex systems theory, which gained increased scientific recognition in the late 1960s. This was inspired by Warren Weaver's 1947 article introducing "Science and Complexity."[8] Ultimately, this theory evolved to embrace four key concepts surrounding the complex adaptive system (CAS): 1) Nature thrives on diversity. 2) A CAS will emerge as a whole from its components, and that whole is typically greater than the sum of its parts. 3) A CAS can self-organize. 4) These features combine to allow the CAS to adapt to its environment. The observations that supported this theory were manifold, crossed disciplines, and could be repeated. Simultaneously, computer systems began to diffuse into society, slowly finding their way into new disciplines while starting to exhibit their own elementary CAS characteristics. Today, the rapidly emerging field of agentic AI harnesses various task-oriented agents to work cooperatively. Compared to the more passive generative AI (GenAI), agentic AI represents a sophisticated automated CAS with the ability to operate autonomously, self-organize, and adapt without prompting. This emergent behavior

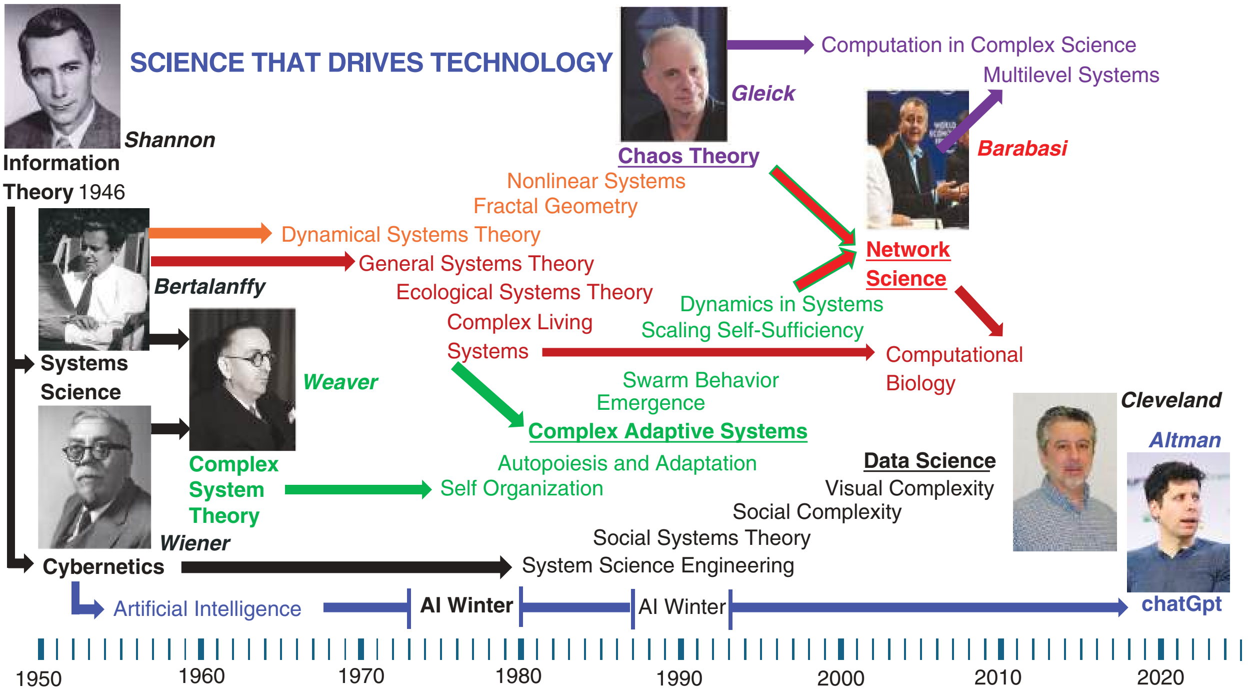


FIGURE 1. Evolution of the sciences that drive modern technology. [Source: Photos from Wikipedia and Purdue University (Cleveland).]

further legitimizes the importance of the 1960s complexity theory as a valid observational science.

Chaos

As the new complexity science gained momentum, systems science further spawned new fields, including nonlinear systems. By the late 1970s, these fields converged with CAS. Chaos theory was born. James Gleick popularized it in his 1987 Pulitzer Prize-winning book, *Chaos: Making New Science*.[9] This theory holds that minute initial conditions, including feedback loops, could lead to catastrophic perturbations in a CAS. Once a CAS reaches the tipping point, it could go chaotically out of control without warning. Ultimately, chaos theory helped refine the notion that a CAS often thrives in a sweet spot between rigidity and the edge of chaos. This suggests a dynamic system could vacate its optimal stable state in either direction based on some seemingly inconsequential input, as theoretically sublime as a butterfly flapping its wings. This concept underscores the need to impose guardrails when building stateful dynamic systems, such as agentic AI.

Together, these forerunning novel sciences became relevant in many disciplines, often blurring distinctions separating disciplines. At the same time, general technology increasingly supports unique, discipline-specific applications.

Network science

In 1737, Leonhard Euler posed the problem of crossing, only once, over each of the seven bridges in Kronenberg.[10] While the problem proved impossible, the solution involved a form of mathematics that led to modern graph theory. Graph theory, once the obscure stuff of graduate-level mathematics, lay dormant until the 1990s. By the late 1990s, it exploded into network science. Albert Lazlo Barabasi popularized the ascendance of network science in his 2002 book, *Linked*.[11] Here, he used the power curve to demonstrate how hubs become predominant while connecting numerous lessor nodes into dynamic, interactive preferential scale-free networks.

While alien to the average citizen, graph-based algorithms have become plentiful and powerful. They serve to quantify CAS behavior. They reinforce statistical data science as popularized by W.S. Cleveland in 2001.[12] They support databases capable of quantifying and visualizing complex relationships. These features underlie Amazon's supply chains, Facebook's social milieu, Google's search acumen, and an array of exceedingly valuable computational routines extending to economy, logistics, business, sociology, astrophysics, neurology, and most traditional disciplines.

More importantly, the ascent of graph algorithms, coupled with the science of dynamic network systems, led to unprecedented levels of interoperability. This is well illustrated by the metaverse, which incorporates blockchain, Internet-of-things, digital twins, edge computing, semantic communication, and federated learning into interoperable digital ecosystems.[13] This level of integration has propelled tech giants to fiscal prominence. Several have quickly grown to be too big to fail. As firms scaled to the cloud, massive data centers began to dot the landscape to warehouse the world's data, ranging from the classics to banal e-mails. As massive social networks evolved, the initial allure of hyperlocal social media connectivity to family and friends has given way to abject marketing and purveyors of misinformation. The Internet has now become a new Wild West, where hucksterism, lies, deepfakes, hallucinations, hate, and violence can too easily be found. All this has evolved while novel graph-based implementations became competitive, closely guarded secrets within the tech giants, never to face public scrutiny, much less profound public understanding.

> Today, science has marched on, but the populace has remained largely unaware of the significance of the dramatic, game-changing discoveries over the ensuing decades.

SCIENCE AND TECHNOLOGY SYMBIOSIS

Figure 2 Traces the emergence of technology at ever-increasing scale and degrees of interoperability since Turing.

Artificial Intelligence

The term *AI*, now omnipresent, was born at Dartmouth University in 1956.[14] In 1965, the first chatbot, Eliza, captured public attention.[15] The same year, Frank Rosenblatt introduced machine learning (ML) and the notion of neurological connection.[16] More than 100 probability-based ML algorithms have since taken hold. As a result, applied probability mathematics and combinatorics have become increasingly crucial as ML algorithms began to scale significantly. By 2002, massive ML training had become a key element in training large language models (LLMs) and converting data centers from data warehouses to massive, energy-consuming, but sub-optimized, ML-based training hubs. LLMs depend upon thousands of GPUs, tensor logic, and vector databases.[17] These developments led to the 2022 show-stopper: GenAI.

GenAI

By 2022, OpenAI CEO, Sam Altman, became a leading spokesman for the revolutionary GenAI. GenAI relies on huge ML-trained data stores, including the Internet. Noam Chomsky's 1953 exploration of syntactic structures[18] prepared the way for natural language processing (NLP). NLP was enhanced by Rumelhart's backpropagation,[19] setting the scene for AI involvement. NLP is now harnessed to mathematically predict the following words in a passage based on the immense LLM as trained by ML.[20] Relying on complex ML and sophisticated NLP, GenAI thrives on forms of mathematics that most budding secondary-school or even undergraduate math students never see. Worse, multilayered probability paths make ML reverse-engineering exceedingly tricky. This is compounded by the Internet, which harbors verifiable facts, copyrighted materials, weighted biases, and outright lies. Nonetheless, LLMs are subject to ingesting everything despite attempts to cleanse "bad" data. Worse, as GenAI is utterly probabilistic, not deterministic, it does not reason. It predicts words without considering their true context. Augmenting GenAI with knowledge graphs helps to contextually temper restricted LLM content.[21] This is a leading strategy to reduce LLM inaccuracy via metadata.

Agentic AI, however, is rapidly eclipsing GenAI. This form of AI embraces CAS principles to behave autonomously. Agentic AI, more oriented to reasoning and adaptive self-organization, raises questions. Who or what design mechanisms maintain the sweet spot between rigidity and chaos in large-scale systems of cooperating agents? Already generating incomprehensible chip designs, what prevents AI from ultimately reproducing itself through access to additive manufacturing?

BELIEF IN MAGIC

Sophisticated popular media bypass deep scientific understanding underpinning today's all-pervasive technology. In early 2025, Y. N. Harari's *Nexus*[22] ranked 14th on the *New York Times* nonfiction best-seller list. Harari, a renowned historian, traces network evolution from the Stone Age through AI. D. B. Auerbach's *Meganets*[23] is a less well-ranked but equally powerful book. Auerbach, a systems designer with years of practical experience, frequently invokes volume, velocity, and virality to explore the loss of control over massive networks, social media, gamification, blockchain, and AI. Both authoritative books examine the influence that massive information age networks, compounded by AI, have on modern society. Harari and Auerbach convincingly defend their main points based on pertinent, compelling, nontechnical, but historically relevant evidence. Harari allows time to engage democratic self-correcting mechanisms to curb runaway automation. Auerbach argues that while some constraints can be applied, the feedback loops in massive networks have already rendered them out of control. Regrettably, however, neither author deeply considers science's impact on the automation dilemmas they chronicle.

The Industrial Age brought linearity, statics, and uniformity. Harari points out that industrialization also resulted in a world war before stabilizing around shared principles. By

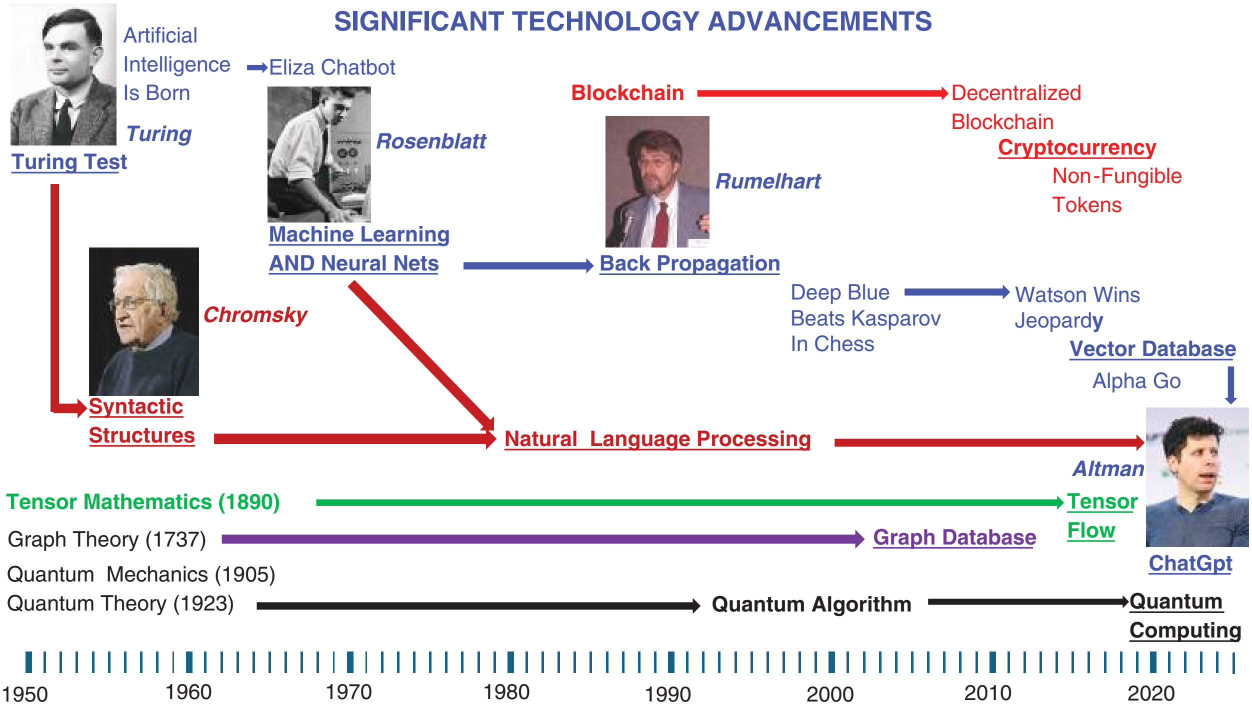


FIGURE 2. Technology advancements developing in parallel with the advancements in science. [Source: Photos from Wikipedia and Cornell University (Rosenblatt).]

contrast, the rapidly evolving information age embraces nonlinear systems, dynamics, revived mathematics, and now, self-organizing and potentially autonomous AI systems. Both authors suggest that technology, where information has overtaken material as the leading commodity, can easily foment social conflict. Harari laments that humans adapt slowly, making enlightened guardrails hard to envision. Nonetheless, the scientific shift has proven both seismic and profound. Public awareness of modern science's fundamental concepts and potential consequences is now essential. Otherwise, any guardrails designed to prevent social collapse from runaway automation or its ruthless exploitation are liable to be ineffective.

ACADEMIC SHORTFALL

Had Harari and Auerbach relied on science in their academically sound works, their respective cases would likely not have been as well-received. Many find today's science exotic: nothing more than pure magic. Many cultural influences, including popular media, reinforce and support a fanciful sublimation of science. Superheroes fly around, conspiracy theories abound, and science denial is socially acceptable. Considered irrelevant, all mathematics takes a back seat for far too many unmotivated TikTok students. Harari notes that media now influences increasing numbers of people at ever faster rates. As such, media can either universally aggrandize counterproductive memes or become highly motivational.

Holistic, influential, ever-changing, and cross-discipline modern technology contradicts former reductionist and deterministic beliefs: the predominant mindsets of industrialization. Instead, new views point to integrated, multidiscipline dynamic systems. Today's tools are dynamic and versatile, no longer uniform, linear, or static. These factors directly challenge the production-line mentality's serial and compartmentalized viewpoint. Nonetheless, this vestigial mentality reinforces existing organizational silos with little or no horizontal mobility. Such conditions also confront the antiquated, strict, discipline-based education as it has existed for decades, despite the growing interdisciplinary symbiosis between modern science and technology.

In this era of disruptive change, perhaps it is high time to break academic silos and boldly celebrate lifelong education based on a holistic understanding of the very technologies we consume to surround and consume us. This includes an appreciation of the science that propels all kinds of modern networks, AI, and the underlying mathematics that empower the information age. Such reform, however, may prove impractical in light of entrenched educational institutionalization and deeply ensconced educator mindsets.

Alternatively, it may be time to harness the abundant streaming multimedia technology in a multifaceted, creative campaign to popularize modern science. Such an initiative could be creatively designed to educate people where they live: on their devices. An interdisciplinary, academia-led leadership consortium would be necessary to maintain independence, objectivity, and veracity. Stony Brook's Alda Center offers a limited exemplar for such a notion (https://www.aldacenter.org). Academic leadership is critical, as elitist tech firms exist to compete and protect their intellectual property while satisfying investors and stockholders.

A multipart streaming series, like Sagan's *Cosmos* series, with a catchy title and style, exceptional production values, and an equally charismatic host as Sagan might serve to kick off such a campaign. Ideally, enlightened investors could be engaged to underwrite such a significant campaign in the name of renaissance.

Failing proactive educational reform of some form, Sagan's warning is already coming to pass. Society will never realize what hit it until we become unwitting digital serfs. Or worse. C

REFERENCES

1. C. Sagan, *The Demon-Haunted World: Science as a Candle in the Dark*. New York, NY, USA: Ballantine Books, 2011.
2. K. Boczkowska, "The homely sublime in space science documentary films: Domesticating the feeling of homelessness in Carl Sagan's cosmos and its sequel," *Kultura Popular*, vol. 54, no. 4, pp. 24–34, 2017.
3. J. Voas and K. Millet, "Data feudalism and e-Personhood — Are we becoming digital serfs?" *Computer*, vol. 3, no. 3, pp. 12–16, Mar. 2025, doi: 10.1109/MC.2024.3513897.
4. C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948, doi: 10.1002/j.1538-7305.1948.tb01338.x.
5. D. Baleanu, V. E. Balas, P. Agarwal, Eds., *Fractional Order Systems and Applications in Engineering*. Cambridge, MA, USA: Academic Press, 2022.
6. N. Weiner, *Cybernetics or Control and Communication in the Animal and the Machine*. Cambridge, MA, USA: MIT Press, 1948.
7. A. Hielronymi, "Understanding systems science: A visual and integrative approach," *Syst. Res. Behav. Sci.*, vol. 30, no. 5, pp. 580–595, Sep. 2013.
8. W. Weaver, "Science and complexity," *Am. Sci.*, vol. 36, no. 4, pp. 536–544, 1948.
9. J. Gleick, *Chaos: Making a New Science*. New York, NY, USA: Penguin, 2008.
10. J. R. Petrella and P. M. Doraiswamy, "From the bridges of Königsberg to the fields of Alzheimer: Connecting the dots," *Neurology*, vol. 80, no. 15, pp. 1360–1362, 2013, doi: 10.1212/WNL.0b013e31828c3062.
11. A. L. Barabási and J. Frangos, *Linked: How Everything is Connected to Everything Else and What It Means for Business, Science, and Everyday Life*. New York, NY, USA: Basic Books, 2014.
12. W. S. Cleveland, "Data science: An action plan for expanding the technical areas of the field of statistics," *Int.*

Stat. Rev., vol. 69, no. 1, pp. 21–6, Apr. 2001, doi: 10.2307/1403527.

13. L. Yang, S. T. Ni, Y. Wang, A. Yu, J. A. Lee, and P. Hui, "Interoperability of the metaverse: A digital ecosystem perspective review," 2024, *arXiv:2403.05205*.
14. J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, "A proposal for the Dartmouth summer research project on artificial intelligence," *AI Mag.*, vol. 27, no. 4, p. 12, Dec. 2006.
15. J. Weizenbaum, "ELIZA—A computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966, doi: 10.1145/365153.365168.
16. F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, Nov. 1958, doi: 10.1037/h0042519.
17. B. Hanindhito and L. K. John, "Accelerating ML workloads using GPU tensor cores: The good, the bad, and the ugly," in *Proc. 15th ACM/SPEC Int. Conf. Perform. Eng.*, May 2024, pp. 178–189, doi: 10.1145/3629526.3653835.
18. N. Chomsky, "Systems of syntactic analysis," *J. Symbolic Logic*, vol. 18, no. 3, pp. 242–256, Sep. 1953, doi: 10.2307/2267409.
19. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.
20. A. Iorliam and J. A. Ingio, "A comparative analysis of generative artificial intelligence tools for natural language processing," *J. Comput. Theor. Appl.*, vol. 1, no. 3, 2024, doi: 10.62411/jcta.9447.
21. W. Fan et al., "A survey on rag meeting LLMs: Towards retrieval-augmented large language models," in *Proc. 30th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2024, pp. 6491–6501, doi: 10.1145/3637528.3671470.
22. Y. N. Harari, *Nexus: A Brief History of Information Networks from the Stone Age to AI*. New York, NY, USA: Random House, 2024.
23. D. B. Auerbach, *Meganets: How Digital Forces Beyond Our Control Commandeer Our Daily Lives and Inner Realities*. London, U.K.: Hachette UK, 2023.

GEORGE F. HURLBURT is a retired federal IT specialist. He serves on the Board of Advisors for the University System of Maryland at Southern Maryland, MD 20619 USA. Contact him at gfhurlburt@gmail.com.

Digital Object Identifier 10.1109/MC.2025.3553356

EDITOR HSIAO-YING LIN
IEEE Member; hsiaoying.lin@gmail.com

How AI Agents Are Transforming Software Engineering and the Future of Product Development

Sriram Panyam, Omlet, Inc.
Praveen Gujar, LinkedIn

Artificial Intelligence (AI) agents are revolutionizing software engineering, boosting productivity while creating new demands for skilled engineers and raising critical ethical challenges.

Software engineering and product development are being transformed by artificial intelligence (AI) agents, automating tasks like code generation and testing, improving productivity while enhancing quality. As AI agents continue to automate routine tasks and boost productivity in software engineering, their impact reaches beyond efficiency gains. This article explores how increased productivity can paradoxically raise demand for engineers while introducing new ethical and workforce challenges.[1] By addressing these issues, leaders can navigate the future of AI in software development. In this article, we will look at some of the risks, what it means to the future of software and product engineering, and how engineering leadership can play a role in paving the way toward this. The key concepts and influences of the factors described in this article can be visualized in Figure 1.

Digital Object Identifier 10.1109/MC.2024.3488378
Date of current version: 28 April 2025

STATE OF AI AGENTS FOR DEVELOPER PRODUCTIVITY AND SOFTWARE ENGINEERING

The integration of artificial intelligence (AI) agents into software engineering workflows has brought significant advancements in the field, particularly in enhancing developer productivity. These AI agents, often designed to automate tasks, assist with code generation, and streamline complex processes, have the potential to reshape how software is engineered. This section will explore the latest state of the art in AI-driven development, focusing on three key observations that highlight the impact and widespread adoption of AI agents within the industry.

As AI agents continue to automate routine tasks and boost productivity in software engineering, their impact reaches beyond efficiency gains.

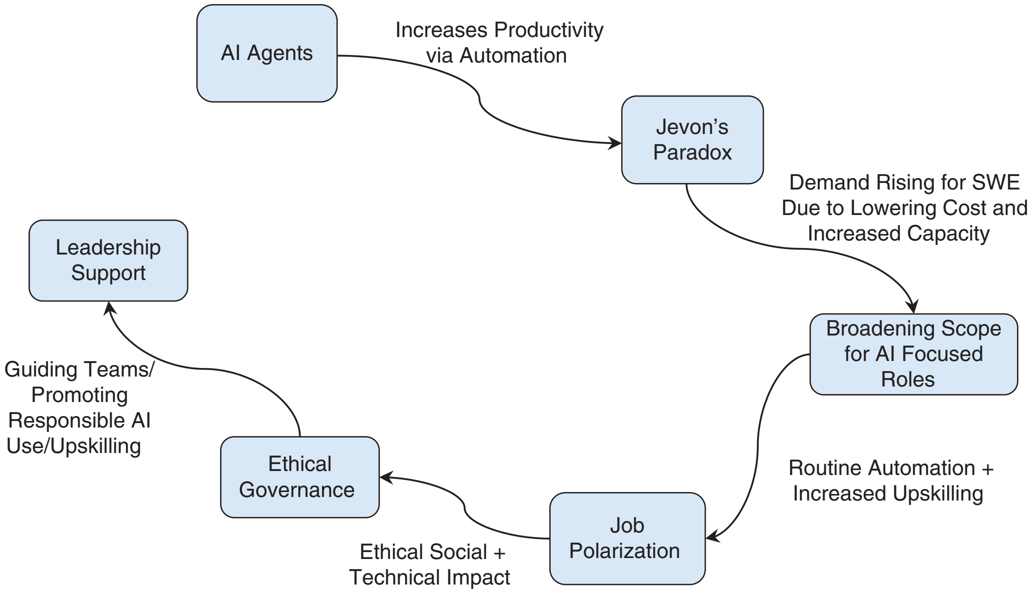
Automated code generation and enhancement

AI's immediate impact on software engineering is evident in automated code generation, exemplified by tools like OpenAI's Codex[2] and GitHub Copilot.[3] These AI models generate code snippets from natural language prompts, streamlining repetitive tasks. By reducing time spent on boilerplate code, they enable developers to focus on more complex, creative work, improving code consistency, and speeding up prototyping. AI agents also offer real-time suggestions, completions, and explanations, reducing cognitive load, and simplifying workflows. As a result, developers can address higher-level challenges, with AI handling routine tasks, significantly boosting productivity and enhancing overall software development efficiency.

AI is significantly transforming software engineering through tools like OpenAI's Codex and GitHub Copilot, which automate code generation and enhance developer productivity. For instance, a controlled experiment demonstrated that developers using GitHub Copilot completed tasks 55.8% faster than those without AI assistance.[4] Moreover, more than 50,000 organizations have adopted GitHub Copilot, with 67% of Accenture developers utilizing it at least five days per week.[5] This widespread adoption underscores AI's role in streamlining repetitive tasks, allowing developers to focus on more complex and creative aspects of software development, thereby enhancing code quality and accelerating project timelines.

Intelligent code reviews and bug detection

AI agents are becoming indispensable in the software development lifecycle, with their applications extending far beyond code generation. For example, AI-driven tools like DeepCode and SonarQube now play pivotal roles in code reviews and bug detection, leveraging machine learning algorithms to identify potential issues. Recent studies indicate that deep learning-based review tools can reduce bug-related incidents by up to 75%,[6] drastically improving software quality. These tools can catch up to 60% of security vulnerabilities that human reviewers might



AI Agents
Increases Productivity via Automation
Jevon's Paradox
Demand Rising for SWE Due to Lowering Cost and Increased Capacity
Broadening Scope for AI Focused Roles
Routine Automation + Increased Upskilling
Job Polarization
Ethical Social + Technical Impact
Ethical Governance
Guiding Teams/ Promoting Responsible AI Use/Upskilling
Leadership Support

FIGURE 1. Interconnected themes in leadership, role scope, and the future of software engineering.

overlook. Integrated into continuous integration/continuous deployment pipelines, they continuously monitor for performance bottlenecks and adherence to coding standards, reducing the need for time-intensive manual reviews. This automation can reduce review times by up to 25%, particularly in large-scale projects.[7] As a result, teams benefit from faster and more reliable deployments, maintaining both high code quality and overall system robustness.

Enhanced collaboration and knowledge sharing

AI agents have also demonstrated the ability to facilitate collaboration among development teams by offering improved knowledge sharing and documentation capabilities. AI-powered tools can automatically generate documentation for complex code, translating raw code into comprehensible explanations. This improves onboarding processes for new team members, allowing them to quickly understand a project's structure and functionality without requiring extensive mentorship from senior developers.[8]

Moreover, AI agents can analyze historical code changes and provide recommendations based on patterns learned from past projects. This enables developers to leverage institutional knowledge even when working across distributed teams or when team members transition out of a project. The ability of AI agents to bridge the gap between documentation and code further enhances developer productivity by reducing communication overhead and improving accessibility to project details.[9]

SO, WHAT IS THE PROBLEM?

The integration of AI agents into software development has led to impressive productivity gains, but it also introduces risks that could reshape the profession. As AI tools take on more complex tasks, engineers may face challenges that mirror historical shifts, like those seen with cloud computing. This section delves into three key risks posed by the growing adoption of AI agents.

Deskilling and erosion of expertise

A major concern is the potential deskilling of engineers. As AI agents handle routine tasks such as coding, debugging, and code reviews, there's a risk that engineers may rely too heavily on these systems, leading to the erosion of critical programming and problem-solving skills. Tasks that once required deep technical understanding are now automated, reducing the need for engineers to engage with the underlying algorithms and technical complexities. Over time, this reliance on AI could result in a workforce less adept at addressing complex issues or understanding the limitations of AI-generated solutions, ultimately weakening the profession's technical depth and expertise.[10] In our own experience onboarding engineers and architects at tech firms, some of our observations are as follows[29,30,31]:

- Automated machine learning platforms like Google AutoML reduce the need for deep data science expertise, making it easier for nonexperts to build models but risking erosion of core modeling skills.
- Copilot's autocompletion and code generation have been reported to make junior developers reliant on AI, potentially diminishing their problem-solving and algorithmic thinking capabilities.
- Platforms (like OutSystems) have enabled nondevelopers to build applications without programming expertise—raising concerns that traditional developers may lose their foundational skills in software architecture and coding logic.

Increased job polarization

AI automation is also likely to polarize the job market, with demand for high-level, strategic roles increasing, while midlevel and entry-level positions diminish. AI agents can now handle many tasks traditionally assigned to junior engineers, such as writing boilerplate code and conducting basic tests. As a result, entry-level roles may shrink, limiting opportunities for new engineers to gain foundational experience.[11] This could lead to a two-tiered workforce, where a small group of highly skilled engineers designs and manages AI systems, while others are left with more repetitive tasks, creating a fragmented career landscape and potentially reducing innovation diversity in the field.[12] From our experience we have found the following[32,33]:

- By automating code generation, Copilot has reduced the need for junior developers to write boilerplate code, potentially shrinking opportunities for entry-level developers.
- Companies using robotic process automation tools have automated routine tasks in sectors like banking and insurance, reducing the demand for junior roles in repetitive process management.
- Automated testing tools like Selenium and AI-driven test platforms have replaced manual quality assurance testers, diminishing entry-level opportunities in quality assurance.

Ethical and accountability challenges

The use of AI in critical engineering tasks raises ethical and accountability concerns. AI-generated code can introduce errors, biases, or security vulnerabilities, complicating responsibility. Engineers may find it difficult to fully understand or control AI-driven decisions, especially when working with opaque machine learning models.[13] This lack of transparency complicates accountability and raises broader questions about ethical deployment, as biased data or outdated practices could perpetuate harmful outcomes without proper oversight.[14] These risks highlight the potential for significant disruption in the software engineering profession, as AI becomes more embedded in the development process. Just some of the issues reported are as follows[34,35,36,37,38]:

- GitHub's AI-powered code assistant was found to suggest biased and vulnerable code snippets due to training on flawed datasets, raising concerns about security and ethical deployment.
- Amazon scrapped its AI hiring tool after it was found to be biased against women, highlighting how flawed training data can perpetuate discrimination in recruitment processes.
- Apple faced allegations of gender bias in its AI credit card algorithm, with female applicants receiving lower credit limits compared to men despite having similar financial profiles.
- Tesla's AI-driven Autopilot faced criticism after a series of accidents, raising accountability questions over the reliance on AI in critical, safety-focused engineering systems.
- Microsoft's AI chatbot, Tay, began spewing racist and inappropriate content after being manipulated by users, highlighting the risks of deploying AI without sufficient safeguards against malicious exploitation.

JEVONS' PARADOX FOR THE FUTURE OF SOFTWARE ENGINEERING

Is the software engineering procession then doomed? Let us explore how these trends might echo Jevons' paradox, and examine whether increased productivity through AI could further reconfigure the profession, akin to the impact of cloud computing.

Jevons' paradox suggests that efficiency gains from technological innovations often lead to increased demand, rather than reduced consumption.[15] In software engineering, AI agents are streamlining development by accelerating code generation, feature delivery, and project scalability. However, instead of reducing the demand for engineers or simplifying software systems, AI-driven automation may increase both, as organizations expand their ambitions and adopt more complex software solutions.

The paradox highlights the risks and opportunities in AI's impact on the profession, particularly concerning deskilling, job polarization, and ethical challenges as follows:

- *Addressing deskilling and expertise erosion*: While the risk of deskilling is real, Jevons' paradox implies that as AI simplifies certain programming tasks, the scope of software engineering will broaden.[16] Engineers will increasingly need to gain expertise in AI-related fields, such as designing and supervising AI agents, creating opportunities for upskilling. Moreover, as AI agents handle routine tasks, engineers can focus on more complex and creative challenges, like system architecture, security, and ethical oversight. In this way, AI may shift engineers' roles toward higher-order tasks that deepen their technical and societal expertise, helping mitigate the risk of expertise erosion.
- *Counteracting job polarization*: Jevons' paradox suggests that AI's increased productivity may counteract job polarization by creating new roles in software engineering. As software systems grow more sophisticated, there will be rising demand for engineers who not only code but also design and manage AI tools. Similar to how the cloud revolution created specialized roles like DevOps and cloud architects, AI could lead to new positions in AI governance, ethical deployment, and integration, expanding opportunities for engineers to specialize.[17]
- *Navigating ethical challenges*: The widespread adoption of AI in software development brings ethical concerns like bias, transparency, and accountability. Jevons' paradox implies that the profession may evolve to address these, requiring engineers to develop ethical literacy and governance structures. As stewards of ethical AI, engineers will need to engage deeply with the societal impacts of their work, ensuring that AI systems are built and deployed responsibly.[18]

SOFTWARE ENGINEERING IN AN AGENTIC WORLD

As AI agents become embedded in software engineering, the profession will undergo a significant transformation, introducing complexities demanding new skills. While AI tools enhance productivity, they will not diminish the need for human engineers. Next are several key areas where this reconfiguration will occur, highlighting the essential role of engineers alongside AI tools:

1. *AI system design and governance*: Software engineers will need to design systems that integrate AI while ensuring these tools function seamlessly with traditional infrastructure. Engineers will also be responsible for implementing governance frameworks to monitor AI performance, detect bias, and ensure ethical usage. This growing demand for oversight underscores the necessity of human expertise in AI-driven systems.[14]
2. *Human–AI collaboration*: AI agents can enhance developers' productivity but cannot replace the nuanced decision-making required in complex software engineering tasks. Engineers will need to collaborate with AI agents, interpreting their outputs and guiding their actions. This human-in-the-loop approach will elevate the need for software engineers to provide context-driven interventions

and corrections, particularly in edge cases or tasks requiring a deeper understanding.[19]

3. *Complex systems integration*: The integration of AI tools into existing development ecosystems will require engineers to balance AI models with traditional software infrastructure. This task will become more complex as engineers must ensure AI scalability, manage data pipelines, and optimize AI–human interactions. Engineers will be crucial in navigating these challenges to ensure the successful deployment of hybrid AI systems.[20]
4. *AI-specific debugging and maintenance*: Debugging AI models presents unique explainability challenges as errors often arise from data or algorithmic assumptions, not traditional code flaws. Engineers will need new diagnostic strategies to address these issues, especially in dynamic environments where AI models are continually retrained.[21]
5. *Ethical engineering and AI accountability*: As AI becomes widespread, engineers will face increasing pressure to ensure ethical, fair, and transparent AI systems. This will involve testing for bias, creating explainability mechanisms, and adhering to legal and regulatory standards, underscoring the necessity of human oversight in AI development.[22]

In short, while AI agents are reshaping the profession, they will also demand new skills and create a more complex engineering landscape that requires deeper human expertise for successful navigation.

ROLE OF THE ENGINEERING LEADER

As AI continues to reshape software engineering, engineering leaders must guide their teams through this transformation. The rise of AI brings both opportunities and challenges, requiring leaders to adopt new strategies for maximizing productivity while managing the increased complexity.

Fostering continuous learning and adaptation

Leaders should encourage a culture of continuous learning, offering AI-specific training and promoting cross-disciplinary collaboration between teams. By fostering adaptability, engineers can seamlessly integrate AI into their workflows and stay updated on emerging AI-driven technologies.[23,24]

Restructuring teams for AI integration

Traditional roles will need to evolve as AI becomes embedded in software development. Leaders must introduce specialized roles, such as AI governance, human–AI collaboration, and ethics oversight, while also creating interdisciplinary teams that merge software engineering with AI expertise.[25]

Balancing automation and human oversight

While AI automates many routine tasks, human oversight remains critical for decision-making and addressing ethical concerns. Leaders should create feedback loops where human engineers validate AI outputs, ensuring automation complements rather than replaces human judgment.[26]

Developing ethical AI frameworks

Engineering leaders must implement ethical AI frameworks to ensure transparency, fairness, and accountability in AI usage. This includes conducting audits, fostering transparency through explainability tools, and ensuring AI aligns with organizational values.[27]

Fostering human–AI collaboration

Leaders must emphasize collaboration between engineers and AI agents, positioning AI as a tool that enhances human creativity. Promoting human-in-the-loop processes and designing AI to augment human skills ensures a productive and ethical integration of AI into the development process.[28]

The integration of AI agents into software engineering offers unprecedented potential for enhancing developer productivity and transforming the profession. While challenges such as deskilling, job polarization, and ethical concerns arise, they also present opportunities for growth, specialization, and innovation. Engineering leaders play a critical role in fostering a productive and responsible AI-driven future, ensuring that AI serves as a tool to amplify human creativity and expertise. By embracing these changes, the software engineering profession can evolve into a more complex yet more capable field, driving a better future for both technology and society.

REFERENCES

1. "France: Adopted German-French recommendations for the use of AI programming assistants." Digital Policy Alert. Accessed: Oct. 7, 2024. [Online]. Available: https://digitalpolicyalert.org/event/23288-adopted-german-french-recommendations-for-the-use-of-ai-programming-assistants
2. *OpenAI Codex*. (Aug. 10, 2021). OpenAI. Accessed: Oct. 7, 2024. [Online]. Available: https://openai.com/index/openai-codex/
3. *GitHub Copilot*. GitHub. Accessed: Oct. 1, 2024. [Online]. Available: https://github.com/features/copilot
4. "Research: Quantifying GitHub copilot's impact in the enterprise with Accenture," *The GitHub Blog*, May 2024. [Online]. Available: https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-in-the-enterprise-with-accenture/
5. "Research: With 12,000 developers using GitHub Copilot, Accenture doubles down on GitHub's platform." GitHub. Accessed: Oct. 7, 2024.

[Online]. Available: https://github.com/customer-stories/accenture

6. J. P. Meher, S. Biswas, and R. Mall, "Deep learning-based software bug classification," *Inf. Softw. Technol.*, vol. 166, Feb. 2024, Art. no. 107350, doi: 10.1016/j.infsof.2023.107350.
7. R. Tufano, O. Dabić, A. Mastropaolo, M. Ciniselli, and G. Bavota, "Code review automation: Strengths and weaknesses of the state of the art," *IEEE Trans. Softw. Eng.*, vol. 50, no. 2, pp. 338–353, Feb. 2024, doi: 10.1109/TSE.2023.3348172.
8. A. Biswal, K. Agarwal, N. Tripathy, S. K. Dewangan, and S. Choubey, "Codesplain: AI powered documentation app," *I-Manager's J. Softw. Eng.*, vol. 18, no. 4, 20–27, 2024.
9. B. Berabi, A. Gronskiy, V. Raychev, G. Sivanrupan, V. Chibotaru, and M. Vechev, "Deepcode AI fix: Fixing security vulnerabilities with large language models," 2024, *arXiv:2402.13291*.
10. S. Bushuyev, D. Bushuiev, V. Bushuieva, N. Bushuyeva, and S. Murzabekova, "The erosion of competencies in managing innovation projects due to the impact of ubiquitous artificial intelligence systems," *Procedia Comput. Sci.*, vol. 231, pp. 403–408, 2024, doi: 10.1016/j.procs.2023.12.225.
11. K. Komp-Leukkunen, "How ChatGPT shapes the future labour market situation of software engineers: A Finnish Delphi study," *Futures*, vol. 160, Jun. 2024, Art. no. 103382, doi: 10.1016/j.futures.2024.103382.
12. S. M. Hyrynsalmi et al., "Bridging gaps, building futures: Advancing software developer diversity and inclusion through future-oriented research," 2024, *arXiv:2404.07142*.
13. N. Rao, "Navigating challenges with LLM-based code generation using software-specific insights," Ph.D. dissertation, Microsoft Research, Redmond, WA, USA, 2024.
14. Q. Lu, L. Zhu, X. Xu, J. Whittle, D. Zowghi, and A. Jacquet, "Responsible AI pattern catalogue: A collection of best practices for AI governance and engineering," *ACM Comput. Surv.*, vol. 56, no. 7, pp. 1–35, 2024, doi: 10.1145/3626234.
15. "Jevon's paradox." Wikipedia. Accessed: Sep 29, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Jevons_paradox
16. F. Rohde et al., "Broadening the perspective for sustainable artificial intelligence: Sustainability criteria and indicators for Artificial Intelligence systems," *Curr. Opin. Environ. Sustainability*, vol. 66, Feb. 2024, Art. no. 101411, doi: 10.1016/j.cosust.2023.101411.
17. J. Sauvola, S. Tarkoma, M. Klemettinen, J. Riekki, and D. Doermann, "Future of software development with generative AI," *Automated Softw. Eng.*, vol. 31, no. 1, 2024, Art. no. 26, doi: 10.1007/s10515-024-00426-z.
18. D. Russo, "Navigating the complexity of generative ai adoption in software engineering," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 5, pp. 1–50, 2024, doi: 10.1145/3652154.
19. D. Tang et al., "Collaborative agents for software engineering," 2024, *arXiv:2402.02172*.
20. M. J. Goswami, "Challenges and solutions in integrating AI with multicloud architectures," vol. 10, no. 10, pp. 68–73, Oct. 2021.
21. S. Cao et al., "A systematic literature review on explainability for machine/deep learning-based software engineering research," 2024, *arXiv:2401.14617*.
22. A. Abusitta, M. Q. Li, and B. C. Fung, "Survey on explainable AI: Techniques, challenges and open issues," *Expert Syst. Appl.*, vol. 255, Dec. 2024, Art. no. 124710, doi: 10.1016/j.eswa.2024.124710.
23. R. Ketrin and Z. Matta, "Developing leadership skills for managing AI teams and projects," pp. 18–23, Jun. 2024, doi: 10.13140/RG.2.2.18792.07688.
24. F. A. Ajayi and C. A. Udeh, "Agile work cultures in IT: A Conceptual analysis of HR's role in fostering innovation supply chain," *Int. J. Manage. Entrepreneurship Res.*, vol. 6, no. 4, pp. 1138–1156, 2024.
25. S. Panyam, P. Gujar, and G. Paliwal, "Evolving product engineering teams in the AI era," *IEEE Eng. Manag. Rev.*, early access, Aug 13, 2024, doi: 10.1109/EMR.2024.3442970.
26. H. Hirvonen and F. Westerling, "Beyond human oversight—Quality management as a tool to control automated decision-making systems," in *The De Gruyter Handbook of Automated Futures*, V. Fors, M. Berg, and M. Brodersen, Eds., Berlin, Germany: De Gruyter, 2024, pp. 255–270.
27. E. Halme, M. Jantunen, V. Vakkuri, K. K. Kemell, and P. Abrahamsson, "Making ethics practical: User stories as a way of implementing ethical consideration in Software Engineering," *Inf. Softw. Technol.*, vol. 167, Mar. 2024, Art. no. 107379, doi: 10.1016/j.infsof.2023.107379.
28. X. Lai and P. L. P. Rau, "AI as coleader: Effects of human consideration and AI structure behaviors on leadership effectiveness and neural activation," *Int. J. Hum.–Comput. Interact.*, pp. 1–19, Jun. 2024, doi: 10.1080/10447318.2024.2365029.
29. "The critical flaw with AutoML: Big problems require human creativity." Accessed: Oct. 15, 2024. [Online]. LinkedIn. Available: https://www.linkedin.com/pulse/critical-flaw-automl-big-problems-require-human-kevin-dewalt
30. Z. Cui, M. Demirer, S. Jaffe, L. Musolff, S. Peng, and T. Salz. "The effects of generative AI on high skilled work: Evidence from three field experiments with software developers." SSRN. Accessed: Oct. 15, 2024. [Online]. Available: https://ssrn.com/abstract=4945566
31. E. Martinez and L. Pfister, "Benefits and limitations of using low-code development to support digitalization in the construction industry," *Autom. Constr.*, vol. 152, Aug. 2023, Art. no. 104909, doi: 10.1016/j.autcon.2023.104909.
32. C. Stokel-Walker. "Why it sucks to be a junior developer right now."

LeadDev. Accessed: Oct. 15, 2024. [Online]. Available: https://leaddev.com/team/why-it-sucks-be-junior-developer-right-now

33. Workbox Technologies. "AI: Transforming the future of manual software testing." Medium. Sep. 2023. Accessed: Oct. 15, 2024. [Online]. Available: https://medium.com/@workboxtech/ai-transforming-the-future-of-manual-software-testing-7b185b00810b
34. R. Wright, "GitHub Copilot replicating vulnerabilities, insecure code," TechTarget, Feb. 23, 2024. [Online]. Available: https://www.techtarget.com/searchsecurity/news/366571117/GitHub-Copilot-replicating-vulnerabilities-insecure-code
35. E. Drage and K. Mackereth, "Does AI debias recruitment? Race, gender, and AI's "eradication of difference," *Philosophy Technol.*, vol. 35, no. 4, 2022, Art. no. 89, doi: 10.1007/s13347-022-00543-1.
36. N. Firth, "Apple Card is being investigated over claims it gives women lower credit limits," MIT Technology Review, Nov. 2019. [Online]. Available: https://www.technologyreview.com/2019/11/11/131983/apple-card-is-being-investigated-over-claims-it-gives-women-lower-credit-limits/
37. A. Jatavallabha, "Tesla's autopilot: Ethics and tragedy," 2024, *arXiv:2409.17380*.
38. M. J. Wolf, K. W. Miller, and F. S. Grodzinsky, "Why we should have seen that coming: Comments on Microsoft's tay "experiment," and wider implications," *ORBIT J.*, vol. 1, no. 2, pp. 1–12, 2017, doi: 10.29297/orbit.v1i2.49.
39. P. Gujar, S. Panyam, and G. Paliwal, "AI integrated product development: Building sustainable competitive advantage," *IEEE Eng. Manag. Rev.*, early access, Oct. 7, 2024, doi: 10.1109/EMR.2024.3475408.

SRIRAM PANYAM is Chief Architect at Omlet, Inc., Sunnyvale, CA 94087 USA. Contact him at sri.panyam@gmail.com.

PRAVEEN GUJAR is a product leader at LinkedIn, Saratoga, CA, 95070 USA. Contact him at praveen.gujar.s@gmail.com.

Digital Object Identifier 10.1109/MC.2025.3553357

Much Ado About DeepSeek …

Michael Zyda, University of Southern California

In this column, I try and make sense of the fear of China's DeepSeek model, as well as reiterate that we all become useless in 2026–2027, if we don't stop the machines.

In Zyda,[1] I pointed out that

> "… the field of computer science [is] pretty global, that our tech papers [are] published and available to anyone who [wants] them. Now they are all on the Internet, so the only thing that can really be blocked during Cold War 2.0 is specialized hardware to China and its surrogates …."[1]

Not software—anyone can replicate software with the right B.S./M.S./Ph.D. in computer science. Also in Zyda:[1]

> "Software and the algorithms underneath cannot be blocked during a cold war. Especially since the country that sends the most graduate students to computer science and electrical engineering programs in the United States is China, with some departments in those fields being 80% to 85% PRC (People's Republic of China) nationals. The only way we can block this technology transfer is to provide permanent visas or citizenship to those students upon graduation so that they stay in the United States. We should do that for anyone from any country who has completed a Masters or Ph.D. in those fields."[1]

Or they will go home to China/wherever and build their own version of that software to great and brilliant acclaim.[8] And that acclaim comes, even if the new version is built on top of something that was previously open source like OpenAI.[9]

It is important to note that the title of this article is in homage to the Shakespeare play, *Much Ado About Nothing* (1598), where according to Wikipedia "*noting*, sounding like "nothing" and meaning gossip, rumor, overhearings)" is exactly the right sentiment for an explication of the DeepSeek state.[4]

Here is what we are going to be "nothing about" (Figure 1).

Digital Object Identifier 10.1109/MC.2025.3541112
Date of current version: 28 April 2025

EDITOR MICHAEL ZYDA
University of Southern California;
zyda@mikezyda.com

DEEPSEEK R1: GOSSIP, RUMOR, OVERHEARINGS ...

Now, the gossip/rumor/overhearings about DeepSeek R1 in this next section are derived from a 30 January 2025 post on Linkedin by Steve Nouri, one of my favorite posters on Linkedin[5] and the DeepSeek FAQ.[2] I have turned Nouri's text into part of the mind map of Figure 1 and then added into the mix other gossip/rumor/overhearings and[2] to complete the "nothing."

In Nouri,[5] Nouri starts out by telling us that the previously unknown-to-us startup DeepSeek had just announced that they had "built a reasoning AI model that competes with OpenAI's best for 1/1000 the cost."[5] This normally wouldn't have caught anyone's attention *except* that it erased US$2 trillion in market cap on NASDAQ, US$500 billion of which was Nvidia's. All of the market lemmings assumed that since the Chinese startup had built it all for just US$6 million that we no longer needed much in the way of Nvidia hardware or much of the OpenAI request for US$500 billion to build their next model. Silicon Valley was streaming tears over this "nothing."

Nouri compared OpenAI's GPT-4 development cost of US$600 million plus training cost against the announcement from DeepSeek that their total cost was US$6 million; that OpenAI's model was not open source and that DeepSeek's model was open source and made from an earlier OpenAI model when it was still open source, according to Vinod Khosla on an X post on the 30 January 2025. So, the sky was falling in Silicon Valley bigly.

So, with DeepSeek saying that it was 27x cheaper than OpenAI's models and cheaper to build and cheaper to run, everyone wanted to hire consultants to ask them the crucial question, should they pull the plug on Silicon Valley's AI efforts or stand pat? Were our guys over here outsmarted by a Chinese development effort? Maybe it was time to stop the Cold War with China over tech transfer before we get locked out!!! DeepSeek is already on Amazon Web Services (AWS) and on Apple's iOS!!! And the sky is still falling!!!

SO, WHAT DID WE ALL FIND OUT?

Well, remember the title of this column is "Much ado about DeepSeek …".

I put out some feelers to friends inside of some of the largest AI companies in Silicon Valley and they all came back with "this is a whole lot of freaking out about nothing. The scaling laws have been predicting that this would be possible for years." So, that is somewhat of an obfuscation. But maybe the only

COMMENTS?

If you have comments about this article, or topics or references I should have cited or you want to rant back to me on why what I say is nonsense, I want to hear. Every time we finish one of these columns, and it goes to print, what I'm going to do is get it up online and maybe point to it at my Facebook (mikezyda) and my LinkedIn (mikezyda) pages so that I can receive comments from you. Maybe we'll react to some of those comments in future columns or online to enlighten you in real time! This is the "Games" column. You have a wonderful day.

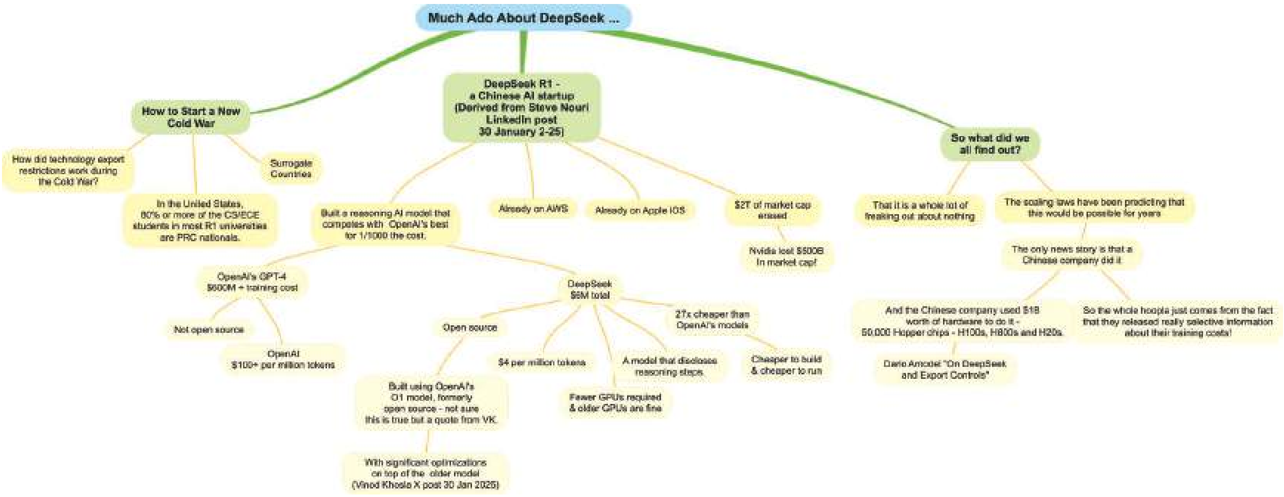

Much Ado About DeepSeek ...
How to Start a New Cold War
How did technology export restrictions work during the Cold War?
In the United States, 80% or more of the CS/ECE students in most R1 universities are PRC nationals.
Surrogate Countries
DeepSeek R1 - a Chinese AI startup (Derived from Steve Nouri LinkedIn post 30 January 2-25)
Built a reasoning AI model that competes with OpenAI's best for 1/1000 the cost.
Already on AWS
Already on Apple IOS
$2T of market cap erased
Nvidia lost $500B In market cap!
OpenAI's GPT-4 $600M + training cost
Not open source
OpenAI $100+ per million tokens
DeepSeek $6M total
Open source
Built using OpenAI's O1 model, formerly open source - not sure this is true but a quote from VK.
With significant optimizations on top of the older model (Vinod Khosla X post 30 Jan 2025)
$4 per million tokens
Fewer GPUs required & older GPUs are fine
A model that discloses reasoning steps
27x cheaper than OpenAI's models
Cheaper to build & cheaper to run
So what did we all find out?
That it is a whole lot of freaking out about nothing
The scaling laws have been predicting that this would be possible for years
The only news story is that a Chinese company did it
And the Chinese company used $1B worth of hardware to do it - 50,000 Hopper chips - H100s, H800s and H20s.
Dario Amodei "On DeepSeek and Export Controls"
So the whole hoopla just comes from the fact that they released really selective information about their training costs!

FIGURE 1. Much ado about DeepSeek ….

news story is that a Chinese company did it. As I started this column, I told you that that kind of software development was pretty likely seeing that we are educating Chinese students here in the United States and they are some of our best students!

There were some other things found out. According to Dario Amodei, DeepSeek did their work using US$1 billion worth of Nvidia hardware, some 50,000 Hopper chips, H100s, H800s, and H20s. Now these are older chips and run slower than the current generation of Nvidia hardware but that US$1 billion has to be added into the DeepSeek development costs.[3]

The only way we can block this technology transfer is to provide permanent visas or citizenship to those students upon graduation so that they stay in the United States.

In Amodei,[3] there are some great summary comments by Amodei on this in his article entitled "On DeepSeek and Export Controls." The three crucial paragraphs are the following:

> "Thus, I think a fair statement is "DeepSeek produced a model close to the performance of U.S. models 7–10 months older, for a good deal less cost (but not anywhere near the ratios people have suggested)".
>
> R1, which is the model that was released last week and which triggered an explosion of public attention (including a ~17% decrease in Nvidia's stock price), is much less interesting from an innovation or engineering perspective than V3. It adds the second phase of training—reinforcement learning, described in #3 of Amodei[3] in the previous section—and essentially replicates what OpenAI has done with o1 (they appear to be at similar scale with similar results).
>
> These will perform better than the multibillion models they were previously planning to train—but they'll still spend multibillions. That number will continue going up, until we reach AI that is smarter than almost all humans at almost all things."[3]

If you are going to use export controls in your title, you are reaching back to locking the barn door after the cows have escaped argument with respect to DeepSeek and Nvidia hardware. But Amodei comes back on this to my point:

> "The performance of DeepSeek does not mean the export controls failed. As I stated above, DeepSeek had a moderate-to-large number of chips, so it's not surprising that they were able to develop and then train a powerful model. They were not substantially more resource-constrained than U.S. AI companies, and the export controls were not the main factor causing them to "innovate." They are simply very talented engineers and show why China is a serious competitor to the US."[3]

Amodei and I agree that Chinese engineers, most likely trained in the United States, are great engineers and capable competitors. And they have 12 times the number of engineers we have in the United States, working at one third the cost of engineers here (my comment from my experience consulting in China[6]).

Yann LeCun, Chief AI Scientist at Meta said the following on Linkedin:[7]

> "To people who see the performance of DeepSeek and think: 'China is surpassing the U.S. in AI.' You are reading this wrong. The correct reading is: 'Open source models are surpassing proprietary ones.'"

So, it's all down to the issue of open source and why globally we all should have that for our critical AI systems, so that we can all just get along ….

WE ALL BECOME USELESS IN 2026–2027

There are some great and thoughtful things in Amodei's paper that I find amazing, this paragraph in particular:

> "Making AI that is smarter than almost all humans at almost all things will require millions of chips, tens of billions of dollars (at least), and is most likely to happen in 2026–2027. DeepSeek's releases don't change this, because they're roughly on the expected cost reduction curve that has always been factored into these calculations."[3]

So, we have rationale to continue to fund the development of these large models and their required large hardware, along with their nuclear power plants so they can run that hardware and its air conditioning and continue the warming of the earth (commentary).

So, in 2026–2027, we will have "AI that is smarter than almost all humans at almost all things." And we all become useless and the AI will write my columns while I swim. Is that a bleak or happy future? Depends on how much you like swimming.

ACKNOWLEDGMENT

The author wishes to thank those readers who have gotten to the end of this bimonthly column without finding all of the deliberate and accidental errors. And I apologize for my personal commentaries but I think they are very important for our futures. And I look forward to being less than fully useless in 2026–2027!

REFERENCES

1. M. Zyda, "Generative AI changes the world, maybe," *Computer*, vol. 57, no. 9, pp. 118–123, Sep. 2024, doi: 10.1109/MC.2024.3416231.
2. "DeepSeek FAQ." Stratechery by Ben Thompson. Accessed: Jan. 30, 2025. [Online]. Available: https://stratechery.com/2025/deepseek-faq/
3. D. Amodei. "On DeepSeek and export controls." darioamodei.com. Accessed: Jan. 30, 2025. [Online]. Available: https://darioamodei.com/on-deepseek-and-export-controls
4. W. Shakespeare. "Much ado about nothing." Wikipedia. Accessed: Feb. 6, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Much_Ado_About_Nothing
5. S. Nouri, "DeepSeek R1 - A Chinese AI startup," *LinkedIn*, Jan. 30, 2025.
6. M. Zyda. *Mike Zyda - Playlists*. (May 3, 2009). [Online]. Available: https://www.youtube.com/@mikezyda/playlists
7. Y. LeCun, *Linkedin*, Jan. 30, 2025.
8. "Understanding DeepSeek's impact on US-China relations," *South China Morning Post*, Feb. 7, 2025. [Online]. Available: https://www.scmp.com/tech/series/3297545/understanding-deepseeks-impact-us-china-relations
9. M. Chen, "Is China's DeepSeek moment a chance to transform into an 'open-source nation'?" *South China Morning Post*, Feb. 3, 2025. [Online]. Available: https://www.scmp.com/news/china/diplomacy/article/3297200/chinas-deepseek-moment-chance-transform-open-source-nation

MICHAEL ZYDA is the founding director of the Computer Science Games Program and a professor emeritus of engineering practice in the Department of Computer Science, University of Southern California, Los Angeles, CA 90089 USA. Contact him at zyda@mikezyda.com.

Digital Object Identifier 10.1109/MC.2025.3553358

Agentic AI and the Cyber Arms Race

Sean Oesch, **Jack Hutchins**, **Phillipe Austria**, and **Amul Chaulagain**, Oak Ridge National Laboratory

In this article, we examine the implications for cyberwarfare and global politics as agentic artificial intelligence becomes more powerful and enables the broad proliferation of capabilities only available to the most well-resourced actors today.

In the early years of cybersecurity, defenders utilized virus-specific signatures, honeypots, and heuristics. As attacks increased in volume and attackers became more sophisticated, moving toward polymorphic malware, packers, and novel evasion techniques, defenders looked to machine learning to provide scalability (quickly analyze large volumes of data and automate repetitive tasks), pattern recognition (detect common attack patterns), and novelty detection (recognize abnormal behaviors that may indicate malicious actors or insider threats). Companies now use large language models (LLMs) to provide analysts and reverse engineers with a rapid analysis of malicious code and best next steps when triaging alerts.[a] But another paradigm shift in cybersecurity for both attackers and defenders is still on the horizon: agentic artificial intelligence (agentic AI).[b]

Cyberwarfare is inherently different from traditional kinetic warfare.[c] Kinetic warfare seeks to force an opponent to submit through physical violence. Cyberwarfare seeks a strategic advantage through espionage, disruption, and degradation of information and operational systems. In cyberwarfare, skill is the weapon and cyberweapons suffer from impermanence—they only work well once, or until the threat is publicly acknowledged and relevant systems patched to prevent the threat vector from being exploited again. As defenders become more sophisticated, the cost of developing an effective cyberweapon

[a] For example, see Microsoft Security Copilot (https://www.microsoft.com/en-us/security/business/ai-machine-learning/microsoft-security-copilot).

[b] This work will be published in *Computer* in the Cybertrust column by IEEE.

[c] See article “Cult of the Cyber Offensive: Misperceptions of the Cyber Offense/Defense Balance” (https://www.yalejournal.org/publications/cult-of-the-cyber-offensive-misperceptions-of-the-cyber-offensedefense-balance).

Digital Object Identifier 10.1109/MC.2025.3544116
Date of current version: 28 April 2025

becomes prohibitively expensive, so that only nation state actors or their equivalents can afford to leverage the skills necessary to create them.

But what if the skills needed to create cyberweapons become widely available through AI agents? While a single agent capable of replacing a skilled human is likely still be a decade or more away,[d] an agent composed of multiple hierarchical models with specific skill sets is immanent. Imagine a centralized reinforcement learning agent (CARL) in control of a suite of task-specific agents: a large reverse engineering model trained to understand, produce, and manipulate binary code, a log agent trained to digest and make inferences from disparate log data, a networking agent capable of mapping and traversing networks, and a vulnerability finder agent able to analyze a system or service and identify effective tactics, techniques, and procedures. Each of these task-specific agents may utilize a multiagent solution as well and interact with existing libraries and tools. CARL is now capable of achieving complex tasks by delegating work to the appropriate task-specific agent, achieving behaviors that mimic those of a skilled cyber operator.

Existing multiagent orchestrations platforms such as CrewAI already allow agents to work together to achieve complex tasks. And several skilled cyber-specific agents already exist. Companies like XBOW and RunSybil[e] use AI agents to automate pentesting, with XBOW able to find and exploit vulnerabilities in 75% of web security benchmarks and discover novel vulnerabilities in web applications.[f] And Dropzone AI uses autonomous agents to automate alert triage and other Tier 1 tasks in the Security Operations Center.

As these cyber agents become more powerful and are combined into multiagent solutions, they have the potential to fundamentally change the nature of and shift the balance of power in the cyber landscape. In this article, we explore the ways that agentic AI may change the symmetry between attackers and defenders based upon the expertise we have developed during our own research.[5,6] We also discuss the potential implications of agentic AI in geopolitics.

IMPLICATIONS FOR THE BALANCE OF POWER IN CYBERWARFARE

Contemporary cybersecurity follows a cyclical pattern where absolute prevention of attacks remains unfeasible; threat actors exploit vulnerabilities, defenders respond with containment measures and patches, and both parties engage in an ongoing process of adaptation and learning. This creates a coevolutionary dynamic between attackers and defenders, each developing increasingly better methods in response to the other's capabilities.[3] We believe the introduction of sophisticated agentic AI into the cyber domain is likely to both shatter and maintain this fundamental pattern. It will maintain this pattern because AI agents for offense and defense possess capabilities for adaptation and evolution in response to one another. When AI attack agents grow in capability, defensive AI agents can adapt through retraining or dynamic adaptive capabilities.

However, agentic AI agents may also shatter the existing paradigm by empowering previously insignificant actors and further exposing entities without the resources to update their own defenses. The cost of maintaining a strong defensive security posture is inherently higher than conducting an attack, especially if that attack is automated, and many organizations already lack the ability to withstand today's threats (see Michael et al.[4] for a discussion of ways AI can empower defenders). Organizations without the money to buy or knowledge to implement effective defenses may be overwhelmed in this new world of agentic AI. Moreover, if AI agents gain the ability to create new offensive attacks in hours, minutes, or even seconds against complex defensive systems, it may not be possible for defensive agents to adapt quickly enough to counter the threats, or defenders may be required to adopt more aggressive strategies, such as adversarial AI (discussed next) or actively finding and compromising adversary infrastructure.

Building on our prior research into adaptive cyber defense agents,[5,6] we have begun to test the ability of attack and defense agents to adapt to one another in a coevolutionary fashion. As can be seen in Figure 1, offensive and defensive AI agents are capable of adapting to improvements in each other's capabilities simply by retraining after the opposing agent is updated.

One key difference between traditional cybersecurity and agentic AI is that AI is vulnerable to unique attacks against its robustness via adversarial AI. Parquini et al.[7] recently demonstrated this difference by tricking an LLM-based AI red agent. When the LLM-based agent attacked, they were able to detect that it was an AI and they changed the system response to make the agent fail. This is only one example of the many ways that AI agents could be compromised and is a vulnerability whether they are being used for offense or defense. Countering adversarial AI and guaranteeing agent robustness will be essential to the future of agentic AI in cyber.

[d] See article from *Our World in Data* (https://ourworldindata.org/ai-timelines).

[e] RunSybil (https://www.runsybil.com/).

[f] How XBOW found a Scoold authentication bypass (https://xbow.com/blog/xbow-scoold-vuln/).

IMPLICATIONS FOR GEOPOLITICS

In addition to impacting the dynamic between offense and defense in cybersecurity, agentic AI has huge potential to shift global power dynamics, much like the advent of nuclear weapons reshaped global power dynamics in the 20th century. In the same way that the Manhattan Project heralded a new era of strategic deterrence, secrecy, and arms racing, the broader accessibility of AI-enabled capabilities could empower not only major powers but also smaller states and even nonstate actors. Unlike nuclear technology, whose development hinged on large scale, highly centralized projects under tight government control, key components of AI research, including open source frameworks and off-the-shelf computing power, may be more diffusely available. This lowers the barrier to entry for countries eager to acquire a new form of deterrence or regional influence, much as smaller states historically leveraged disruptive military technologies to counterbalance conventional power asymmetries.[8]

The result could be a two-tiered ecosystem, reminiscent of early atomic history, in which a handful of powerful actors retain access to the most cutting-edge models requiring immense data resources and sophisticated infrastructure, while mid-level states capitalize on "good-enough" AI to carry out disruptive operations. Much like the nuclear era ultimately extended beyond the initial superpowers despite attempts at containment,[1] it is unlikely that the most advanced AI capabilities will remain exclusive for long. Once foundational knowledge and baseline technologies become widely understood, ambitious states can funnel resources, both overtly and covertly, into domestic AI labs or forge alliances to acquire expertise. The historical diffusion of strategic technologies underscores how determined nations eventually bridge initial capability gaps, especially when regional ambitions or security dilemmas drive them forward.[8]

Smaller states that perfect even moderately sophisticated autonomous cyber operations could punch above their weight. A well-placed AI-driven intrusion could degrade vital infrastructure, extract sensitive data, or sow panic by manipulating information systems at key moments. This ability to project power in cyberspace mirrors how early nuclear programs granted regional players disproportionate diplomatic leverage.[2] In a volatile geopolitical environment, autonomous cyber systems could serve as tools for smaller nations to disrupt or deter larger powers, achieving strategic objectives without the risks and costs associated with conventional military engagements (see Wingfield et al.[9] for examples of how a cyber campaign can impact warfare).

From a geopolitical standpoint, this leveling effect is tempered by the tendency of leading powers to maintain an edge. Historically, strategic advantages in technology have often created a lag period during which dominant players consolidate their position before broader proliferation occurs. In cyberspace, however, the pace of innovation and the decentralized nature of AI development could compress this timeline, potentially resulting in rapid horizontal proliferation of mid-tier capabilities. Meanwhile, vertical proliferation (the constant refinement of top-tier AI systems by technologically advanced nations) will likely exacerbate global inequalities, creating an arms race dynamic that pushes smaller players to adopt asymmetrical tactics.

Unlike the nuclear age, where deterrence mechanisms and mutual assured destruction eventually stabilized conflict between superpowers, agentic AI introduces greater unpredictability. The opacity and speed of cyber operations complicate attribution, raising the likelihood of retaliatory actions based on suspicion rather than certainty. Without the transparency and

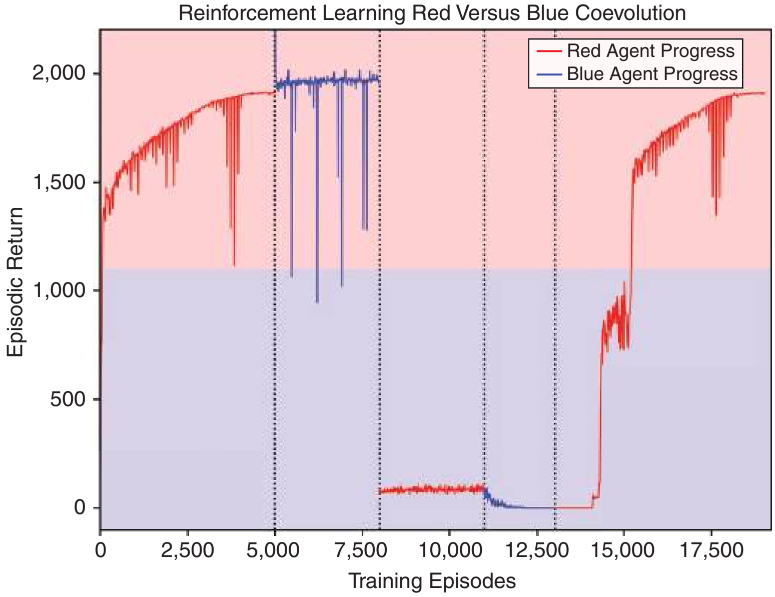


FIGURE 1. Graph showing AI Red/Blue Agent coevolution. A low episodic return indicates the blue agent is performing well, while a high episodic return indicates the red agent is performing well. The vertical lines represent different training runs. In the first run, the red agent is trained without a blue agent. Next, the blue agent is trained against this version of the red agent, and so on. As can be seen, over multiple runs the agents can learn to adapt to changes in one another's abilities, effectively coevolving. Cyberwheel, the environment that generated this graph, is on ORNL's Github.

verifiability that characterized Cold War-era arms treaties, the digital domain may foster an arms race with few constraints. Historical parallels to the "long peace" of the Cold War[2] suggest that stability relied on a balance of power and clear communication—factors that are notably absent in cyberspace, where attacks often unfold covertly and instantaneously.

One critical divergence from past strategic technologies lies in the role of smaller states. With AI, the cost of entry is lower, and the need for traditional industrial capabilities is reduced. This dynamic positions agentic AI as a potential equalizer, enabling less resourced nations to exert influence in their regions. Such states may use autonomous cyber tools to project power, deter aggression, or destabilize adversaries. This aligns with historical patterns of smaller nations leveraging disruptive technologies for strategic advantage, yet agentic AI's versatility and speed could amplify these effects beyond what was possible with previous tools like ballistic missiles or drones.

Together, these developments suggest a future in which global order becomes more fluid and unpredictable than under the bipolar stability of the Cold War, or the unipolar stability that followed. Mutual assured destruction hinged on transparent demonstrations of nuclear potency, codified by treaties and shaped by crises that reinforced a balance of terror. In contrast, agentic AI developments and deployments evolve in secrecy, often without the oversight or accountability necessary to prevent escalation. History has shown that disruptive military technologies are rarely contained indefinitely,[1] and agentic AI appears poised to follow this trajectory. Whether its proliferation yields more frequent low-level cyber skirmishes or destabilizing conflicts among major powers remains uncertain. However, what is clear is that autonomous cyber capabilities will augment the arsenals of dominant players while empowering smaller and emerging states to assert themselves in ways that echo, and may also eclipse, the transformations wrought by nuclear weapons in the 20th century.

ACKNOWLEDGMENT

This manuscript has been authored by UT-Battelle, LLC under Contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes. The U.S. Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). This manuscript was prepared as part of the Emerging and Cyber Security Technologies initiative at Oak Ridge National Laboratory. C

REFERENCES

1. W. Burr and J. T. Richelson, "Whether to "strangle the baby in the cradle": The United States and the Chinese nuclear program, 1960-64," *Int. Secur.*, vol. 25, no. 3, pp. 54–99, 2000, doi: 10.1162/016228800560525.
2. J. L. Gaddis, "The long peace: Elements of stability in the postwar international system," *Int. Secur.*, vol. 10, no. 4, pp. 99–142, 1986, doi: 10.2307/2538951.
3. W. Hoffman, "AI and the future of cyber competition," Center for Security and Emerging Technology, Washington, DC, USA, 2021. [Online]. Available: https://cset.georgetown.edu/publication/ai-and-the-future-of-cyber-competition/
4. J. B. Michael and T. C. Wingfield, "Defensive AI: The future is yesterday," *Computer*, vol. 54, no. 9, pp. 90–96, Sep. 2021, doi: 10.1109/MC.2021.3092480.
5. S. Oesch et al., "The path to autonomous cyber defense," 2024, *arXiv:2404.10788*.
6. S. Oesch et al., "Towards a high fidelity training environment for autonomous cyber defense agents," in *Proc. 17th Cyber Secur. Experimentation Test Workshop*, 2024, pp. 91–99, doi: 10.1145/3675741.3675752.
7. D. Pasquini, E. M. Kornaropoulos and G. Ateniese, "Hacking back the AI-hacker: Prompt injection as a defense against LLM-driven cyberattacks," 2024, *arXiv:2410.20911*.
8. J. Schmid, "The determinants of military technology innovation and diffusion," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, USA, 2018.
9. T. C. Wingfield and J. B. Michael, "Waterfall: Cascading effects of a strategic cyber campaign," *Computer*, vol. 56, no. 4, pp. 143–148, Apr. 2023, doi: 10.1109/MC.2023.3244019.

SEAN OESCH is a researcher at Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA. Contact him at oeschts@ornl.gov.

JACK HUTCHINS is a researcher at Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA. Contact him at hutchinsjr@ornl.gov.

PHILLIPE AUSTRIA is a researcher at Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA. Contact him at austriaps@ornl.gov.

AMUL CHAULAGAIN is a software engineer at Oak Ridge National Laboratory, Oak Ridge, TN 37830 USA. Contact him at chaulagaina@ornl.gov.

©ISTOCKPHOTO.COM/SITTIPONG PHOKAWATTANA

Agentic Artificial Intelligence for Cyber Threat Management

Nir Kshetri, The University of North Carolina at Greensboro
Jeffrey Voas, IEEE Fellow

This article explores how agentic artificial intelligence enhances cybersecurity by autonomously identifying and responding to evolving threats. It also addresses potential misuse and vulnerabilities, emphasizing the need for effective safeguards and oversight to mitigate risks.

In 2024, the global artificial intelligence (AI) in cybersecurity market was valued at US$24.8 billion. It is expected to grow to US$29.64 billion in 2025 and exceed US$146.5 billion by 2034.[1] Agentic AI, representing the third wave of AI, is expected to account for a growing share of this market. Agentic AI systems can autonomously detect, respond to, and mitigate security and fraud threats in near real time.[2] AI agents enable personalized protocols that adapt to specific threats and vulnerabilities.[2] A network of specialized models, each tasked with identifying different vulnerabilities, share insights and collectively address threats.[3] This capability reduces response times to potential attacks, enhancing overall security by providing faster, more efficient threat management.[2] AI is soon expected to shift from chatbots to agent-driven cybersecurity, enhancing threat detection, autonomous responses, scalability, and cyber hygiene.[4] Given the global shortage of nearly 4 million cybersecurity professionals,[5] agentic AI has the potential to fill this gap by automating tasks, improving efficiency, and enhancing security operations.

On the other side of the coin, agentic AI can also bring increased security risks. Unlike traditional AI systems that function within controlled environments, AI agents interact with various systems and external data sources, expanding the potential attack surface. This can lead to

DISCLAIMER

The authors are completely responsible for the content of this article. The views expressed here are their own.

Digital Object Identifier 10.1109/MC.2025.3544797
Date of current version: 28 April 2025

unauthorized access, data leakage, and other vulnerabilities. Weak integration or system flaws have previously allowed sensitive data to be exposed, while malicious actors or coding errors can manipulate AI agents, causing unintended disruptions or financial losses.[6] If an agentic AI system is hacked, the consequences can be severe. First, detecting and confirming the breach may take time, and even a minor change can lead to significant effects. Second, the system's autonomy introduces new security vulnerabilities that must be addressed. Finally, the greatest risk lies in deploying a system without proper monitoring, logging, and controls; these cannot be added as afterthoughts.[7]

This article examines the role of agentic AI in transforming cybersecurity by enabling autonomous threat detection, response, and mitigation. It also highlights the security risks and potential malicious use of agentic AI, emphasizing the importance of robust implementation and oversight.

ENHANCING CYBERSECURITY WITH AGENTIC AI

Agentic AI is revolutionizing security operations centers (SOCs) by automating decision making and adapting to evolving threats.[8] Agentic AI enhances workflow efficiency by automating tasks, like alert triage and incident response, with applications in cybersecurity for autonomous threat detection and response.[9] Leading cybersecurity providers have started offering agentic AI solutions to enhance efficiency and strengthen security(Table 1).[8]

ReliaQuest's GreyMatter platform

ReliaQuest claimed its autonomous AI security agent, launched in September 2024, processes security alerts 20 times faster than traditional methods while improving threat detection accuracy by 30%. This advancement could significantly enhance cybersecurity response times and efficiency.[10] The autonomous AI agent for security operations automates 98% of security alerts and reduces threat containment time to under 5 min. ReliaQuest claims its AI agent, trained on over a decade of incident response data, autonomously handles tier 1 and tier 2 security tasks by analyzing alerts and executing necessary actions, using real-time, customer-specific data in a secure, private environment to eliminate AI hallucinations.[11]

CrowdStrike Falcon Agent

Designed to integrate various advanced endpoint protection features, the CrowdStrike Falcon Agent—integral to the CrowdStrike Falcon platform—operates with a streamlined, lightweight architecture under 20 MB.[12] The platform, driven by AI and the CrowdStrike Security Cloud, monitors attack indicators, threat intelligence, and telemetry data to deliver accurate threat identification, automated protection and remediation, advanced threat hunting, and vulnerability prioritization.[13] CrowdStrike's Charlotte AI Detection Triage, launched in February 2025, is reported to triage security detections with more than 98% accuracy, eliminating more than 40 h of manual work per week, and enhancing SOC operations and response times.[13]

Twine's "digital employee" Alex

Twine, a Tel Aviv, Israel-based cybersecurity startup, aims to address the talent shortage by developing AI agents like Alex, its first "digital employee," specializing in identity and access management. In this role, Alex performs the task of identifying vulnerabilities and taking proactive steps to block unauthorized access, reducing the burden on IT and cybersecurity teams.[14] Deployed as a software-as-a-service platform, Alex connects to customer systems, creates task plans, seeks approval, and executes them with full visibility. In November 2024, Twine received US$12 million in seed funding. Twine is the first company to pioneer cybersecurity digital employees, building on earlier models like Amelia by IPsoft.[15]

TABLE 1. Some AI agents and their cybersecurity performance.

Company	AI agent/platform	Cybersecurity performance
CrowdStrike	Charlotte AI Detection Triage	Delivers more than 98% accuracy in security detection triage, automating more than 40 h of manual work weekly to improve SOC operations and speed up threat responses.
ReliaQuest	GreyMatter platform	Processes security alerts 20 times faster than traditional methods. Automates 98% of security alerts. Reduces threat containment time to under five minutes.
Twine	"Digital employee" Alex	Identifies vulnerabilities and proactively prevents unauthorized access, easing the workload for IT and cybersecurity team.

In addition to advancements in cybersecurity, agentic AI can also enhance software development by streamlining processes throughout the software development lifecycle (SDLC). Integrating intelligent agents into the SDLC enables organizations to shift from reactive to proactive AppSec practices. These AI-powered agents continuously analyze code repositories for vulnerabilities, using techniques like static code analysis and machine learning. What sets agentic AI apart is its ability to adapt to each application's unique structure, building a comprehensive code property graph (CPG) to identify security flaws based on their impact, rather than relying on generic severity ratings.[16] A CPG is a flexible, language-neutral representation of program code that supports scalable, incremental, and distributed analysis.[17] The most intriguing use of agentic AI in AppSec is automated vulnerability fixing. Traditionally, human programmers manually identify and fix vulnerabilities, a process that is time-consuming and error-prone. With agentic AI and CPG analysis, AI agents can detect vulnerabilities and generate context-aware, nonbreaking fixes that address security flaws without introducing new bugs.[16]

Given the global shortage of nearly 4 million cybersecurity professionals, agentic AI has the potential to fill this gap by automating tasks, improving efficiency, and enhancing security operations.

SECURITY AND PRIVACY RISKS OF AI AGENTS

While AI agents offer new capabilities, they also introduce additional risks. As AI agents grow more advanced and interconnected, they may introduce additional security vulnerabilities and potential data leaks. If not carefully implemented, agentic AI could pose privacy and cybersecurity risks to sensitive cloud data.[4] First, unlike standard AI models, where threats are limited to inputs, processing, outputs, and software vulnerabilities, AI agents expand the attack surface to include the entire chain of interactions they initiate, many of which remain invisible to human or system operators.[18] For instance, companies may face challenges with "shadow AI," where employees deploy agentic AI tools without proper authorization. This can lead to significant governance and security risks.[4] These risks include data exposure, unauthorized coding logic errors leading to breaches, and supply chain vulnerabilities from third-party libraries.[18] Manipulation by external threats, including competitors, hackers, or cybercriminals, is a major risk to agentic AI systems. Such interference can reduce algorithm accuracy or cause it to go rogue, leading to financial losses and reputational damage. Even with a small role in a department's operations, a cyberattack could have severe consequences. A typical 2% failure rate might be manageable, but if it spikes to 100%, the resulting downtime could trigger expensive, large-scale failures.[19]

The growing adoption of multiagent systems can create new attack vectors and vulnerabilities if not secured from the outset. Threats like data poisoning, prompt injection, and social engineering—already affecting single-agent systems—could pose even greater risks in multiagent environments due to their expanded network of connections and interfaces, amplifying potential impacts.[4]

Second, these risks are heightened as AI agents process large volumes of personal data. For example, agents managing emails or investment decisions may handle sensitive personal or financial information, increasing a company's vulnerability to cyberattacks.[20]

Third, AI agents, due to their role as gateways to sensitive data and critical systems, present unique risks beyond those associated with typical AI applications. For instance, if a logistics agent were compromised, it could spread malicious commands throughout a supply chain, leading to significant and widespread disruption.[6]

Fourth, the dynamic nature of AI agents makes effective threat detection challenging. These risks are further amplified when models are connected to other systems, potentially increasing vulnerabilities.[6]

THE MALICIOUS POTENTIAL OF AGENTIC AI IN CYBERCRIME

Cybercriminals are also reported to be leveraging this technology, such as self-improving phishing campaigns. Malwarebytes predicts that agentic AI could dramatically enhance cybercriminal tactics by automating and strategizing attacks.[10] While generative AI tools have previously increased attack efficiency, agentic AI could mark a significant shift, allowing attackers to independently reason, plan, and execute more sophisticated operations.[10]

Malicious AI could autonomously identify vulnerabilities and adapt strategies in real time, making attacks more sophisticated and harder to defend against.[10] AI agents learn from past attacks to refine future ones, perform automated spear phishing, adapt in real time to the target's response, and execute multistage attacks. Additionally, agentic AI can combine multiple communication channels, like text and deepfake calls, to improve the effectiveness of phishing attempts.[21] Malwarebytes' 2025 State of Malware report argues that advancements in agentic AI could enable ransomware gangs to automate attacks, targeting multiple victims simultaneously.[10]

Compromising just one component in an AI system can allow lateral movement, enabling cybercriminals to corrupt algorithms, exfiltrate sensitive data, or gather intelligence for future attacks. If the data feeding agentic AI, whether external or internal, is poisoned, it can significantly affect its behavior. The introduction of inaccurate or biased data can increase hallucinations, amplify existing biases, or impair decision making. This can lead to the AI behaving in unexpected ways, and without human oversight, these actions might go unnoticed for some time.[19]

Considering the potential misuse discussed earlier, cybercriminals are testing autonomous AI agents for attacks, but their unreliability prevents full-scale use. Experts anticipate these systems will require significant fine-tuning before becoming effective.[10]

MITIGATING RISKS AND STRENGTHENING GOVERNANCE FOR AGENTIC AI INTEGRATION

Proactive business and IT leaders will embrace agentic AI while mitigating risks by setting clear guardrails, enforcing strict data-access policies, and promoting organizational best practices.[4] It is essential to understand that AI agents will operate in organizations like human employees, interacting with other agents and integrating into human resources systems with distinct permissions and access. These agents will require onboarding and offboarding, making them susceptible to attacks and necessitating robust AI governance and security measures. Having noted the potential for criminal use, cybercriminals are experimenting with AI-driven attacks, although experts suggest refinement is needed before they can be fully effective.[4]

A governance framework that aligns with national and global standards is critical for mitigating cybersecurity risks tied to agentic AI. These policies ensure that implementation and use remain ethical, transparent, and secure, addressing autonomy-related threats. Security professionals must also consider the level of agency granted to the system, its complexity, and the environment in which it operates, as these factors significantly impact its potential security risks.[19] Implementing threat modeling, least-privilege access, and separating trusted from untrusted zones can reduce vulnerabilities. Additionally, monitoring agent activity and maintaining audit trails help detect anomalies effectively.[6]

There is also the need for employee training and automated detection systems to mitigate these risks.[4] Enhancing security awareness through dynamic training tools tailored to risk levels is essential. Organizations should also educate employees about social engineering risks and foster a culture of cybersecurity.[21] To mitigate AI agents' potential threats, IT leaders should educate their organizations on AI agent risks, detect and flag anomalous activities, and map all AI agent interactions to ensure compliance with enterprise policies.[18]

To combat advanced social engineering—an increasingly prevalent threat[22]—organizations can leverage agentic AI to monitor and detect threats, analyze behavioral patterns, and prioritize vulnerabilities for faster response. As AI evolves, employees must be prepared for autonomous agents in the workplace, and businesses should deploy their own AI-based security agents to stay ahead of cybercriminals.[21]

Agentic AI represents a transformative advancement in cybersecurity by enhancing threat detection, autonomous response, and operational efficiency. Companies like ReliaQuest, CrowdStrike, and Twine are already leveraging this technology to streamline SOC tasks, improve accuracy, and reduce response times. Moreover, agentic AI's ability to continuously analyze and adapt throughout the SDLC offers proactive protection against vulnerabilities. However, these benefits are accompanied by significant risks, such as expanded attack surfaces, data leaks, and potential manipulation by cybercriminals. The dynamic nature of multiagent systems and their interactions with sensitive data further amplifies security challenges. As adoption grows, robust monitoring, governance, and risk-mitigation strategies are crucial to maximize agentic AI's potential while minimizing vulnerabilities. Balancing innovation with cybersecurity resilience will be essential for navigating the future of agent-driven solutions.

REFERENCES

1. Precedence Research, "Artificial Intelligence (AI) in cybersecurity market size, share, and trends 2024 to 2034," Nov. 13, 2024. Accessed: Feb. 19, 2025. [Online]. Available: https://www.precedenceresearch.com/artificial-intelligence-in-cybersecurity-market
2. G. Gross, "Agentic AI: 6 promising use cases for business," *CIO*, Nov. 14, 2024. Accessed: Feb. 19, 2025. [Online]. Available https://www.cio.com/article/3603856/agentic-ai-promising-use-cases-for-business.html
3. "Agentic AI: Institute for experiential AI position," Northeastern University Institute for Experiential AI, Jan. 28, 2025. Accessed: Feb. 19, 2025. https://ai.northeastern.edu/news/agentic-ai-institute-for-experiential-ai-position
4. S. Weigand, "Cybersecurity in 2025: Agentic AI to change enterprise security and business operations in year ahead," *SC Media*, Jan. 9, 2025. Accessed: Feb. 19, 2025. [Online]. Available: https://www.scworld.com/feature/ai-to-change-enterprise-security-and-business-operations-in-2025
5. "Strategic cybersecurity talent framework," World Economic Forum, Cologny, Switzerland. Apr. 28, 2024.

Accessed: Feb. 16, 2025. [Online]. Available: https://www.weforum.org/publications/strategic-cybersecurity-talent-framework/

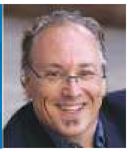
6. R. Ramesh, "What enterprises need to know about agentic AI risks: Mitigating cybersecurity, privacy risks for new class of autonomous agents," *BankInfoSecurity*, Jan. 13, 2025. Accessed: Feb. 19, 2025. [Online]. Available: https://www.bankinfosecurity.com/what-enterprises-need-to-know-about-agentic-ai-risks-a-27282
7. S. Kaufman, "Beyond ChatGPT: The rise of agentic AI and its implications for security," *CSO*, Oct. 22, 2024. Accessed: Feb. 19, 2025. [Online]. Available: https://www.csoonline.com/article/3574697/beyond-chatgpt-the-rise-of-agentic-ai-and-its-implications-for-security.html
8. L. Columbus, "Cybersecurity at AI speed: How agentic AI is supercharging SOC teams in 2025," *VentureBeat*, Jan. 13, 2025. Accessed: Feb. 19, 2025. [Online]. Available: https://venturebeat.com/ai/cybersecurity-at-ai-speed-agentic-ai-supercharging-soc/
9. E. Lisowski, "AI agents vs agentic AI: What's the difference and why does it matter?" *Medium*, Dec. 18, 2024. Accessed: Dec. 29, 2024. [Online]. Available https://medium.com/@elisowski/ai-agents-vs-agentic-ai-whats-the-difference-and-why-does-it-matter-03159ee8c2b4#:~:text=Both%20AI%20Agents%20and%20Agentic%20AI%20are%20changing%20the%20world,experiences%2C%20and%20solving%20complex%20problems
10. S. Klappholz, "Agentic AI could be a blessing and a curse for cybersecurity," *ITPro*, Feb. 5, 2025. [Online]. Accessed: Feb. 19, 2025. Available: https://www.itpro.com/security/cyber-crime/agentic-ai-cybersecurity-risks
11. "ReliaQuest launches first autonomous, self-learning AI agent for security operations," *ReliaQuest*, Sep. 26, 2024. Accessed: Feb. 19, 2025. [Online]. Available: https://www.reliaquest.com/news-and-press/reliaquest-launches-first-autonomous-self-learning-ai-agent-for-security-operations/
12. "CrowdStrike falcon agent: A radical new approach proven to stop breaches." Cosive. Accessed: Feb. 13, 2025. [Online]. Available: https://www.cosive.com/capabilities/crowdstrike-falcon-agent#:~:text=The%20CrowdStrike%20Falcon%C2%AE%20platform%20is%20built%20on,efficacy%20against%20a%20wide%20variety%20of%20threats.&text=A%20single%20lightweight%20agent%20works%20everywhere%2C%20including,providing%20protection%20even%20when%20endpoints%20are%20offline
13. CrowdStrike, "CrowdStrike delivers the next breakthrough in AI-powered agentic cybersecurity with Charlotte AI detection triage," *Business Wire*, Feb. 13, 2025. Accessed: Feb. 19, 2025. [Online]. Available: https://www.businesswire.com/news/home/20250212935384/en/CrowdStrike-Delivers-the-Next-Breakthrough-in-AI-Powered-Agentic-Cybersecurity-with-Charlotte-AI-Detection-Triage
14. D. Riley, "AI meets cybersecurity: Twine launches with $12M funding for digital cyber employees," *SecurityWeek*, Nov. 20, 2024. Accessed: Feb. 19, 2025. [Online]. Available: https://siliconangle.com/2024/11/20/ai-meets-cybersecurity-twine-launches-12m-funding-digital-cyber-employees/
15. G. Press, "This AI agent will defend you from cyber attacks," *Forbes*, Nov. 20, 2024. Accessed: Feb. 19, 2025. [Online]. Available: https://www.forbes.com/sites/gilpress/2024/11/20/this-ai-agent-will-defend-you-from-cyber-attacks/
16. C. Guerrero, "The power of agentic AI: How autonomous agents are revolutionizing cybersecurity and application security," *Dev*, Feb. 4, 2025. Accessed: Feb. 19, 2025. [Online]. Available: https://dev.to/friendgrass7/the-power-of-agentic-ai-how-autonomous-agents-are-revolutionizing-cybersecurity-and-application-h31
17. "Code property graph - specification and tooling [GitHub repository]." GitHub. Accessed: Feb. 9, 2025. [Online]. Available: https://github.com/ShiftLeftSecurity/codepropertygraph
18. H. Vella, "Agentic AI set to rise, with new cybersecurity risks: Gartner," *AI Business*, Dec. 2, 2024. Accessed: Feb. 19, 2025. [Online]. Available: https://aibusiness.com/automation/agentic-ai-set-to-rise-with-new-cybersecurity-risks-gartner
19. "Developing security protocols for agentic AI applications," *Security Boulevard*, Jan. 22, 2025. Accessed: Feb. 19, 2025. [Online]. Available: https://securityboulevard.com/2025/01/developing-security-protocols-for-agentic-ai-applications/
20. Dorsey & Whitney LLP, "Launching agentic AI in an uncertain U.S. regulatory landscape," *JD Supra*, Jan. 29, 2025. Accessed: Feb. 19, 2025. [Online]. Available: https://www.jdsupra.com/legalnews/launching-agentic-ai-in-an-uncertain-u-2524341
21. S. Sjouwerman, "How agentic AI will be weaponized for social engineering attacks," *SecurityWeek*, Feb. 5, 2025. Accessed: Feb. 19, 2025. [Online]. Available: https://www.securityweek.com/how-agentic-ai-will-be-weaponized-for-social-engineering-attacks
22. N. Kshetri, *The Quest to Cyber Superiority: Cybersecurity Regulations, Frameworks, and Strategies of Major Economies*. Cham, Switzerland: Springer-Verlag, 2016.

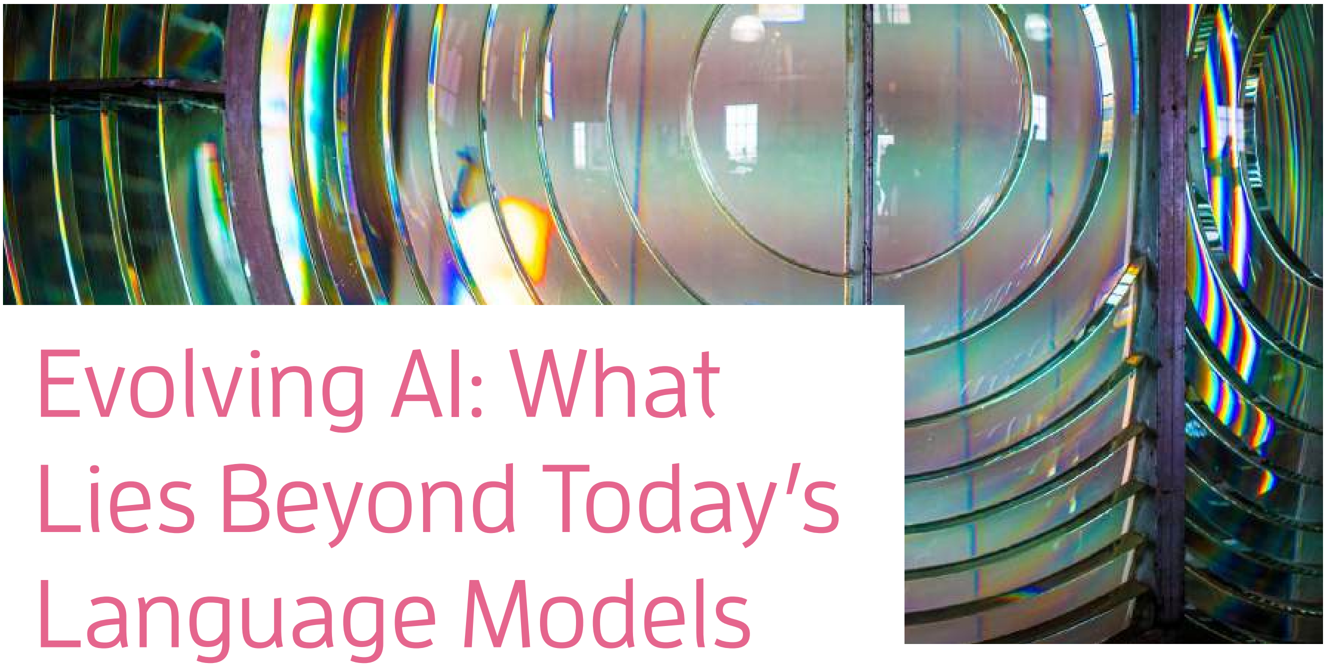
NIR KSHETRI is a professor at the Bryan School of Business and Economics, University of North Carolina at Greensboro, Greensboro, NC 27412 USA. Contact him at nbkshetr@uncg.edu.

JEFFREY VOAS, Gaithersburg, MD 20899 USA, is the editor in chief of *Computer*. He is a Fellow of IEEE. Contact him at j.voas@ieee.org.

IT INNOVATION

EDITOR MARK CAMPBELL
3dot Insights; mark@3dotinsights.com





Evolving AI: What Lies Beyond Today's Language Models

Mlađan Jovanović, Singidunum University
Mark Campbell, 3dot Insights LLC

As language models evolve, a convergent adaptive model approach promises to build on current trends to overcome existing limitations and create a more human-like experience; however, some challenges remain.

Transformer-based language models (LMs) have transformed a broad spectrum of applications by enabling the creation of machine intelligence that can computationally model and interact with the real world, its relationships, and processes without major disruption.[1] While LMs have shown significant advancements in replicating human-like reasoning, such as resolving pronoun ambiguities (Winograd schema problems),[2] they are more effective in constrained areas like math and logic than in unstructured general reasoning scenarios.

LMs

LMs are now in widespread use in software development as customizable coding copilots, and their use as a standard software architecture component and design pattern is becoming commonplace.[3] However, the efficacy of an LM software development copilot still relies heavily on the proficiency of a skilled human programming pilot. Developing artificial intelligence (AI) systems that can leverage generalized reasoning and tacit knowledge as effectively as humans remains a key area for research.

AI EVOLUTION

According to Kuhn's seminal work *The Structure of Scientific Revolutions*, scientific progress is not a linear process but a sequence of revolutions transitioning from the current paradigm to a new one caused by the accumulation of problems that the current paradigm cannot solve. A new paradigm then emerges and stabilizes over time.[4] Like all scientific paradigms, AI continues to evolve from narrowly defined domains to general-purpose applications (as shown in Figure 1).

AI passed through five distinct phases as it transitioned from high to low human supervision as follows:

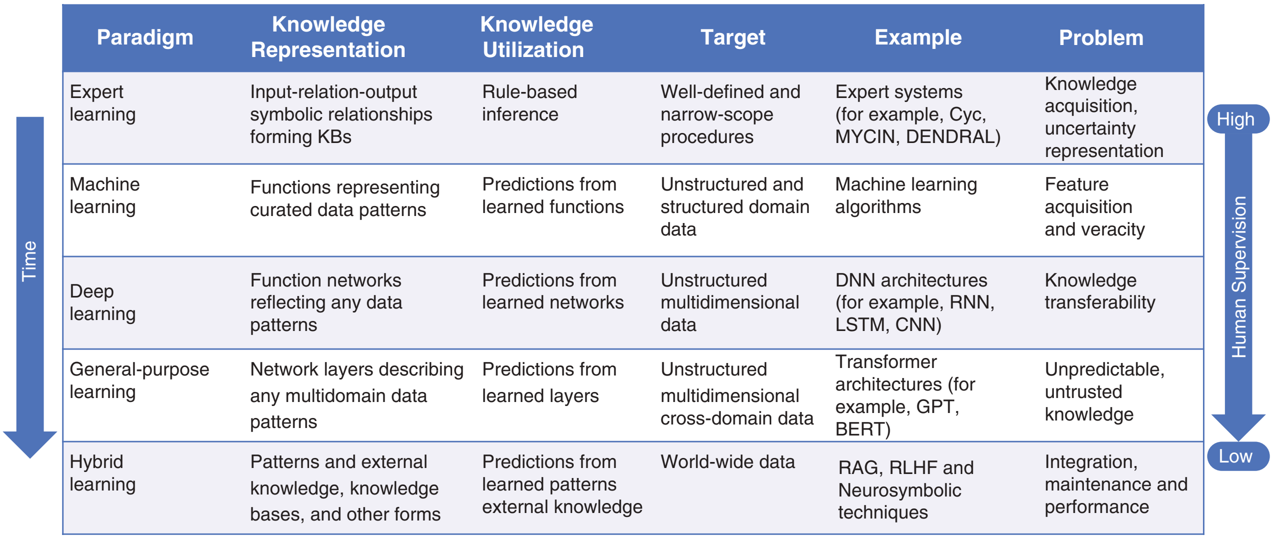
Digital Object Identifier 10.1109/MC.2025.3546045
Date of current version: 28 April 2025

- *Expert learning:* Expert learning is an early AI approach that focused on capturing human expertise in specific domains by codifying expert knowledge into a set of rules and using predefined inference tools to apply these rules to specific tasks. However, this approach proved inadequate for acquiring and maintaining reliable knowledge and applying it effectively to the uncertainties of the real world.
- *Machine learning:* Machine learning (ML) creates models by analyzing large amounts of raw data and using statistical methods to handle uncertainty in the data. However, ML models often lack transparency and require human oversight for tasks such as feature selection and parameter tuning.
- *Deep learning:* Deep learning (DL) reduces the need for human input by directly processing diverse types of data. DL models are able to approximate nearly any function and improve with satisfactory performance. However, they increasingly contain large amounts of complex hidden knowledge, making it difficult to transfer or apply them to other domains.
- *General-purpose learning:* General-purpose learning processes a wider variety of data types in much larger quantities to create pretrained and fine-tuned LMs, such as large LMs, capable of working across different domains. However, these models are expensive, resource intensive, and prone to producing inaccurate, harmful, or biased content.
- *Hybrid learning:* Hybrid learning combines LMs with external knowledge sources using either manual methods—such as Reinforcement Learning from Human Feedback (RLHF), neurosymbolic learning, or integration with knowledge bases (KBs)—or automatic methods—such as Retrieval Augmented Generation (RAG) with vector databases. However, this approach comes at the cost of reduced system precision, increased response time (latency), more complex deployment processes, and higher maintenance requirements.

LM LIMITATIONS

Despite significant advancements in AI, today's most sophisticated LMs still face several critical limitations, including the following:

- *Uncertainty and accuracy:* LMs process all input tokens with equal confidence based on learned statistical patterns. Their performance is evaluated on prediction confidence rather than factual accuracy, and consequently, they lack self-awareness of their knowledge boundaries. This lack of uncertainty qualification can result in undue trust in LM output and faulty decision making by human users.[5]
- *Ethical constraints:* Current systems still struggle to consistently apply ethical constraints across diverse scenarios, often failing to navigate complex moral dilemmas or adapt ethical principles to nuanced real-world situations without human oversight. This poses significant challenges in fields like higher education, where LMs can be misused for plagiarism, potentially compromising students' acquisition of


Time
High
Human Supervision
Low

Paradigm	Knowledge Representation	Knowledge Utilization	Target	Example	Problem
Expert learning	Input-relation-output symbolic relationships forming KBs	Rule-based inference	Well-defined and narrow-scope procedures	Expert systems (for example, Cyc, MYCIN, DENDRAL)	Knowledge acquisition, uncertainty representation
Machine learning	Functions representing curated data patterns	Predictions from learned functions	Unstructured and structured domain data	Machine learning algorithms	Feature acquisition and veracity
Deep learning	Function networks reflecting any data patterns	Predictions from learned networks	Unstructured multidimensional data	DNN architectures (for example, RNN, LSTM, CNN)	Knowledge transferability
General-purpose learning	Network layers describing any multidomain data patterns	Predictions from learned layers	Unstructured multidimensional cross-domain data	Transformer architectures (for example, GPT, BERT)	Unpredictable, untrusted knowledge
Hybrid learning	Patterns and external knowledge, knowledge bases, and other forms	Predictions from learned patterns external knowledge	World-wide data	RAG, RLHF and Neurosymbolic techniques	Integration, maintenance and performance

FIGURE 1. AI evolution. DNN: deep neural network; CNN: convolutional neural network; RNN: recurrent neural network; KB: knowledge base; RAG: retrieval augmented generation; RLHF: reinforcement learning from human feedback; LSTM: long short-term memory; DENDRAL: Dendritic Algorithm; BERT: bidirectional encoder representations from transformers.

essential knowledge and critical thinking skills.[6]

- *Task adaptation*: While LMs excel in various applications, they struggle with certain fundamental tasks. For instance, multiple LMs have failed to perform basic naive time-series forecasting methods, such as predicting values over a period or at a specific future point.[7] Despite extensive training and computational resources, their text-based sequence modeling does not translate well to numerical time-series data, as evidenced by their unchanged performance when input data are randomly shuffled.
- *Safety concerns:* LMs are often fine-tuned for safety, aiming to produce outputs deemed harmless by human standards. However, this safety alignment can be circumvented through carefully crafted inputs designed to elicit harmful responses such as jailbreak attacks or adversarial prompts.[8] Recent research indicates that even safety-aligned LMs remain vulnerable to such attacks, particularly when attackers leverage available model information such as training details to construct adaptive and unsafe requests.
- *Lack of transparency:* Recent research has improved our ability to explain LM decisions by linking their internal mechanisms, such as attention matrices, to specific domain knowledge, particularly in clinical settings.[9] However, significant challenges persist. Creating and maintaining comprehensive and up-to-date KBs for specific domains remains crucial to maintain overall model performance and explainability.

LMs on their own are still far from achieving human-level intelligence. As a result, researchers are actively exploring various alternatives to enhance and complement their capabilities.

EMERGING MODEL TRENDS AND APPROACHES

Looking ahead, current trends suggest the emergence of several promising fields aimed at addressing the existing limitations of LMs.

Neurosymbolic AI

As LMs scale, their generalizability, robustness, and transparency offer diminishing returns. The Neurosymbolic approach presents a promising solution by integrating symbols, logic, and knowledge to guide DL processes toward more efficient, consistent, and explainable outcomes.[10] This approach introduces symbolic constraints as ontologies for logical outputs, uses symbolic reasoning for verifiable results aligned with known information, infuses knowledge graphs for context-sensitive content generation, and incorporates regulatory and ethical norms to ensure compliance. Additionally, it emphasizes the development of comprehensive metrics for better evaluating LMs.[11]

RL

Traditional RL requires prior knowledge of the target environment, limiting its application in real-world scenarios where such knowledge is often incomplete. Newer RL techniques like MuZero overcome this by learning an environment model and using tree-based search, achieving expert-level performance in strategic games like Go, shogi, and chess without predefined rules.[12] RLHF aligns LMs with human values and preferences and is critical in optimizing sequence-specific objectives that are not easily differentiable in supervised fine-tuning.[13] RLHF does, however, face scalability challenges due to the high cost and inconsistency of human feedback. To address this, RL from AI Feedback (RLAIF) uses AI models for feedback instead of humans, although it still relies on preference labels from existing LMs that may carry biases.[14] These advancements aim to enhance RL's adaptability to complex environments while improving alignment with human preferences.

Synthesis of Tailored Architectures

DL model architectures encounter optimization challenges due to limited search spaces and simplistic heuristics. The Synthesis of Tailored Architectures (STAR) approach offers a novel solution by representing models as hierarchical numeric sequences, or genomes, which evolve through gradient-free evolutionary optimization guided by objectives like parameter count, cache size, perplexity, and latency.[15,16] These genomes are encoded into linear input-varying systems, functioning as flexible neural computational units. STAR integrates various components, including attention mechanisms, recurrent structures, and convolutional layers, resulting in improved performance on downstream tasks compared to traditional Transformer architectures.[15]

Mixture of Universals

Mixture of Universals (MoU) enhances time-series forecasting by addressing both short-term and long-term dependencies. Its Mixture of Feature Extractors (MoF) improves the representation of time-series patches for short-term dynamics, while Mixture of Architectures (MoA) integrates Mamba, Feedforward, Convolution, and Self-Attention architectures to model long-term dependencies effectively.[17]

Transformer-Mamba Mixture of Experts

The Transformer-Mamba Mixture of Experts is a hybrid architecture combining Transformer and Mamba models and offers increased context length and lower memory usage. Its quantized version supports cost-effective inference and outperforms open-weight models on long-context benchmarks. These advancements reflect ongoing efforts to improve the efficiency and effectiveness of LMs and time-series forecasting techniques.[18]

Liquid neural model types

Liquid Time-Constant Networks (LTCs) are advanced recurrent neural networks that dynamically adjust their response speed based on input data.[19] This adaptability allows them to effectively handle real-world scenarios with fluctuating data, such as time-series prediction tasks with distribution shifts.[20] Building on this concept, Liquid Foundation Models (LFMs) are a family of generative LTCs designed to model diverse sequential data types, including time series, signals, text, audio, and video.[21] Liquid neural models represent a significant advancement in handling complex time-varying data across multiple domains.

State space models

Structure State Spaces (SSSs) are a modeling approach from control theory that effectively addresses long-range dependencies in sequences.[22] State space models (SSMs) structure neural networks' hidden states as a state machine, using time-dependent matrix representations to predict successive states based on current states and inputs.[22] They map continuous input sequences to latent states and derive predicted output sequences. Although SSMs initially struggled with language modeling performance and hardware efficiency compared to attention-based models, recent advancements have improved these aspects.[23] Selective SSSs, such as Mamba,[24] optimize parameters based on selected input data for better computational efficiency. These models now compete with or outperform Transformer models across various domains[25] and benchmarks,[26] including image[27] and video[28] processing tasks. This evolution highlights the potential of SSSs and SSMs as effective alternatives to traditional attention-based approaches.

Learnable activation models

Kolmogorov-Arnold networks (KANs) introduce a novel approach to neural network architecture by learning activation functions on the edges rather than using predefined node activation functions, with each weight parameter represented by a univariate spline function.[29] This design enhances accuracy, transparency, and scaling efficiency compared to traditional multilayer perceptrons (MLPs), particularly in theoretical contexts[30] and specific domains like mathematics, physics,[29] and scientific discovery.[31] The Kolmogorov-Arnold Transformer (KAT) builds on this concept by replacing MLPs with KANs, demonstrating improved performance while addressing challenges related to scalability, inference time, parameter computation, and weight initialization.[32] However, the effectiveness of KAT in domains beyond vision is still awaiting verification, indicating the potential for KANs to transform neural network design across various fields.

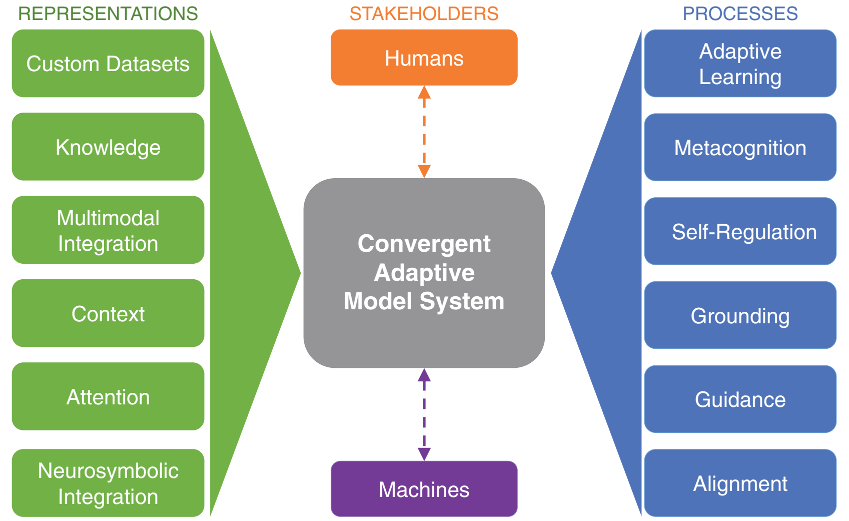


FIGURE 2. Convergent Adaptive Model road map.

THE CONVERGENT ADAPTIVE MODEL APPROACH— A PROPOSED ROAD MAP

We propose the Convergent Adaptive Model approach to extrapolate current AI model trends into a composite model architecture. The proposed road map (Figure 2) illustrates the representations, stakeholders, and processes involved in transitioning from today's goal-oriented AI systems to a more Convergent Adaptive Model approach.

Representations

The *Convergent Adaptive Model* approach relies on *custom datasets* and reliable and structured *knowledge* as its foundation. By implementing *multimodal integration*, it combines various data types such as text, images, audio, and sensor input, mimicking how biological intelligence synthesizes information from multiple sources, enabling a more comprehensive understanding of the world. The system processes individual data points while also grasping their *context*, including relationships and broader implications within their specific domains. *Attention* mechanisms play a crucial role in efficiently allocating resources, focusing on the most pertinent information across different modalities to prioritize and process critical data. *Neurosymbolic integration* merges raw data representations with symbolic and semantic abstractions, allowing the AI to operate seamlessly across multiple levels of abstraction.

Processes

Adaptive learning enables AI systems to continuously evolve in dynamic environments, complemented by *metacognition* for self-observation and behavior adjustment. *Self-regulation*, driven by environmental and user feedback, facilitates ongoing refinement and increased autonomy. For user acceptance and adoption, three critical elements are essential: *grounding*, which ensures accurate matching between AI representations and real-world entities; *guidance*, providing timely and contextual instructions reflecting intentions; and *alignment* with human values and expectations. These components collectively create a Convergent Adaptive Model system that is adaptable, self-improving, and more closely aligned with human needs and values.

CONVERGENT ADAPTIVE MODEL CHALLENGES

While the proposed Convergent Adaptive Model approach offers significant benefits, several obstacles need to be addressed before it can be widely adopted. These challenges include the following:

- *Trust:* Trust is a significant barrier to the widespread adoption of AI applications in various human activities. This trust deficit arises from the lack of transparency in LMs, making it difficult for users to understand their decision-making processes. Additionally, LMs pretrained on noncurated data may contain noise, bias, and false information, leading to unpredictability and misalignment with human goals. While variability in LM responses can indicate creativity, users generally expect consistent and reliable behavior from technology. These factors contribute to user hesitance to fully embrace new technologies, underscoring the need for improved transparency, reliability, and alignment in Convergent Adaptive Model systems.
- *Fairness:* Fairness is a crucial concern in the development of LMs, particularly regarding their alignment with ethical norms and the protection of users' dignity and privacy.[33] Interdisciplinary studies have proposed design principles to identify and mitigate potential harms in ML and DL systems, especially in areas like computer vision.[33] For Convergent Adaptive Model systems to be effective, they must successfully integrate these fairness practices across various domains, addressing the unique challenges and ethical considerations that arise in different fields.
- *Benchmarking:* Benchmarking plays a crucial role in assessing the progress of LMs[34] by comparing their performance against established standards.[35] However, transparency in the process leading to these advancements is equally important.[36] This transparency should reveal whether improvements stem from innovative training methods or from extensive computational resources during inference, such as exhaustive searches exploring numerous alternative solutions.[36] By providing clarity on the sources of progress, researchers and users can better understand the true nature of LM advancements and their practical implications.
- *Unlearning:* Unlearning is a significant challenge for LMs because their learned information is distributed across multiple neurons and layers. This fragmented knowledge makes it difficult to isolate and remove specific concepts as residual traces may persist even after deletion from one neuron.[37] Convergent Adaptive Model systems must effectively unlearn ambiguous, outdated, private, and sensitive information while retaining updated knowledge and maintaining performance. This capability is essential for adapting to new information, respecting privacy concerns, and ensuring accuracy without being hindered by obsolete or problematic data.
- *Randomness:* Randomness significantly impacts real-world decision making, where the dynamics and effects of choices are often unclear. Sources of randomness are hard to quantify and can change over time, complicating accurate modeling. Unaccounted factors can greatly influence outcomes, making it difficult to automate decision processes, especially in low-data scenarios with sequential decisions that affect one another. As a result, replacing human decision makers with a Convergent Adaptive Model system remains a considerable challenge due to the unpredictable nature of real-world environments.

AI systems have evolved from early expert systems to advanced hybrid learning models, yet challenges persist. As new model types and techniques continue to emerge, the proposed Convergent Adaptive Model approach aims to further enhance this evolution by creating integrated systems that better mimic human intelligence. While this approach seeks to address current limitations through improved data integration and adaptive learning, significant challenges related to trust, fairness, unlearning, and handling real-world randomness remain. Overcoming these hurdles is essential for unlocking the full potential of AI across various domains. C

REFERENCES

1. H. A. Simon, *The Sciences of the Artificial*, 3rd ed. Cambridge, MA, USA: MIT Press, 1996.
2. V. Kocijan, E. Davis, T. Lukasiewicz, G. Marcus, and L. Morgenstern, “The defeat of the Winograd Schema Challenge,” *Artif. Intell.*, vol. 325, Dec. 2023, Art. no. 103971, doi: 10.1016/j.artint.2023.103971.
3. Q. Lu, L. Zhu, X. Xu, Z. Xing, and J. Whittle, “Toward responsible AI in the era of generative AI: A reference architecture for designing foundation model-based systems,” *IEEE Softw.*, vol. 41, no. 6, pp. 91–100, Nov./Dec. 2024, doi: 10.1109/MS.2024.3406333.
4. T. S. Kuhn, *The Structure of Scientific Revolutions*, 3rd ed. Chicago , IL, USA: Univ. Chicago Press, 1996.

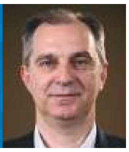
5. T. G. J. Rudner and H. Toner, "Key concepts in AI safety: Reliable uncertainty quantification in machine learning," Center for Security and Emerging Technology, Washington DC, USA, Jun. 2024. [Online]. Available: https://cset.georgetown.edu/wp-content/uploads/CSET-Key-Concepts-in-AI-Safety-Reliable-Uncertainty-Quantification-in-Machine-Learning.pdf
6. B. Borges et al., "Could ChatGPT get an engineering degree? Evaluating higher education vulnerability to AI assistants," *Proc. Nat. Acad. Sci.*, vol. 121, no. 49, 2024, Art. no. e2414955121, doi: 10.1073/pnas.2414955121.
7. M. Tan, M. A. Merrill, V. Gupta, T. Althoff, and T. Hartvigsen, "Are language models actually useful for time series forecasting?" in *Proc. 38th Annu. Conf. Neural Inf. Process. Syst.*, 2024.
8. M. Andriushchenko, C. Francesco, and N. Flammarion, "Jailbreaking leading safety-aligned LLMs with simple adaptive attacks," 2024, *arXiv:2404.02151*.
9. S. Dalal, D. Tilwani, M. Gaur, S. Jain, V. L. Shalin, and A. P. Sheth, "A cross attention approach to diagnostic explainability using clinical practice guidelines for depression," *IEEE J. Biomed. Health Inform.*, vol. 29, no. 2, pp. 1333–1342, Feb. 2025, doi: 10.1109/JBHI.2024.3483577.
10. A. Sheth, and K. Roy, "Neurosymbolic value-inspired artificial intelligence (why, what, and how)," *IEEE Intell. Syst.*, vol. 39, no. 1, pp. 5–11, Jan./Feb. 2024, doi: 10.1109/MIS.2023.3344353.
11. M. Gaur and A. Sheth, "Building trustworthy NeuroSymbolic AI systems: Consistency, reliability, explainability, and safety," *AI Mag.*, vol. 54, no. 1, pp. 139–155, 2024.
12. J. Schrittwieser et al., "Mastering Atari, Go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020, doi: 10.1038/s41586-020-03051-4.
13. L. Ouyang et al., "Training language models to follow instructions with human feedback," 2022, *arXiv:2203.02155*.
14. H. Lee et al., "RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback," 2024, *arXiv:2309.00267*.
15. A. W. Thomas, R. Parnichkun, A. Amini, S. Massaroli, and M. Poli, "STAR: Synthesis of tailored architectures," 2024, *arXiv:2411.17800*.
16. C. Franzen. "Liquid AI's new STAR model architecture outshines transformer efficiency." VentureBeat. Accessed: Jan. 15, 2025. [Online]. Available: https://venturebeat.com/ai/liquid-ais-new-star-model-architecture-outshines-transformer-efficiency/
17. S. Peng, Y. Xiong, Y. Zhu, and Z. Shen, "Mamba or transformer for time series forecasting? Mixture of Universals (MoU) is all you need," 2024, *arXiv:2408.15997*.
18. B. Lenz et al., "Jamba-1.5: Hybrid transformer-mamba models at scale," 2024, *arXiv:2408.12570*.
19. R. Hasani, M. Lechner, A. Amini, D. Rus, and R. Grosu, "Liquid time-constant networks," in *Proc. 38th AAAI Conf. Artif. Intell.*, 2024, vol. 35, no. 9, pp. 7657–7666.
20. M. Chahine et al., "Robust flight navigation out of distribution with liquid neural networks," *Sci. Rob.*, vol. 8, no. 77, pp. 1–14, 2023, doi: 10.1126/scirobotics.adc8892.
21. "Liquid foundation models: Our first series of generative AI models." Liquid.AI. Accessed: Jan. 15, 2025. [Online]. Available: https://www.liquid.ai/liquid-foundation-models
22. A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," 2021, *arXiv:2111.00396*.
23. D. Y. Fu, T. Dao, K. K. Saab, A. W. Thomas, A. Rudra, and C. Ré, "Hungry hungry hippos: Towards language modeling with state space models," 2022, *arXiv:2212.14052*.
24. A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," 2023, *arXiv:2312.00752*.
25. H. Qu et al., "A survey of mamba," 2024, *arXiv:2408.01129*.
26. R. Waleffe et al., "An empirical study of mamba-based language models," 2024, *arXiv:2406.07887*.
27. L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," 2024, *arXiv:2401.09417*.
28. K. Li et al., "VideoMamba: State space model for efficient video understanding," 2024, *arXiv:2403.06977*.
29. Z. Liu et al., "KAN: Kolmogorov-Arnold networks," 2024, *arXiv:2404.19756*.
30. Y. Wang, J. W. Siegel, Z. Liu, and T. Y. Hou, "On the expressiveness and spectral bias of KANs," 2024, *arXiv:2410.01803*.
31. Z. Liu, P. Ma, Y. Wang, W. Matusik, and M. Tegmark, "KAN 2.0: Kolmogorov-Arnold networks meet science," 2024, *arXiv:2408.10205*.
32. X. Yang and X. Wang, "Kolmogorov-Arnold transformer," 2024, *arXiv:2409.10594*.
33. T. Gebru and R. Denton, "Beyond fairness in computer vision: A holistic approach to mitigating harms and fostering community-rooted computer vision research," *Found. Trends® Comput. Graph. Vis.*, vol. 16, no. 3, pp. 215–321, 2024, doi: 10.1561/0600000102.
34. M. Knoop, and F. Chollet. "Abstraction and reasoning corpus prize." ARC Prize. Accessed: Jan. 15, 2025. [Online]. Available: https://arcprize.org/
35. F. Chollet, "On the measure of intelligence," 2019, *arXiv:1911.01547*.
36. M. Mitchell. "Did OpenAI just solve abstract reasoning?" AI: A Guide for Thinking Humans. Accessed: Jan. 15, 2025. [Online]. Available: https://aiguide.substack.com/p/did-openai-just-solve-abstract-reasoning
37. A. F. Cooper et al., "Machine unlearning doesn't do what you think: Lessons for generative AI policy, research, and practice," 2024, *arXiv:2412.06966*.

MLAĐAN JOVANOVIĆ is an associate professor at Singidunum University, 11000 Belgrade, Serbia. Contact him at mjovanovic@singidunum.ac.rs.

MARK CAMPBELL is the founder of 3dot Insights, Sedalia, CO 80135 USA. Contact him at mark@3dotinsights.com.

CYBER-PHYSICAL SYSTEMS

EDITOR DIMITRIOS SERPANOS
CTI DIOPHANTUS and University of Patras;
serpanos@computer.org



Interoperability: A Key to the Future

Nikolina Renieri, Computer Technology Institute and Press DIOPHANTUS

Dimitrios Serpanos, Computer Technology Institute and Press DIOPHANTUS and University of Patras

Interoperability is key to the provision of value-added services in the increasing digitization of governments, organizations, and processes. It requires a renewed, multilayer approach to address challenges and cope with risks that span from correctness to privacy.

INTEROPERABILITY: A GROWING NEED

Interoperability is the ability of different systems, devices, applications, or organizations to work together and exchange information in a seamless, efficient, and meaningful way. It ensures that these components can communicate, interpret, and use shared data without compatibility issues, regardless of their differences in design, technology, or origin. Interoperability is a need in various fields such as e-government, health care, smart cities, etc., to reduce redundant tasks and manual intervention, enable automated workflows, promote better decision making based on shared and accurate information, improve users' experience, and encourage innovation by allowing diverse technologies and vendors to collaborate.

Interoperability can be viewed as the evolution and integration of properties and functionality, such as compatibility, connectivity, internetworking, and distributed services. These features have been addressed since the beginning of computing but are becoming increasingly combined and more complex as IT and operational technology (OT) systems of different organizations are integrating around the world, exploiting heterogeneous computing and networking technologies.

Considering the evolution of systems and services, efforts to address challenges have evolved through the years, from the development of hardware compatibility frameworks to open systems interconnection models to an approach

Digital Object Identifier 10.1109/MC.2025.3547072
Date of current version: 28 April 2025

for levels of conceptual interoperability.[1] However, efforts have been focusing either on specific levels of system/service abstraction, such as hardware or distributed systems, and specific domains, such as environments with digital twins,[2] or, most importantly, on specific computational, networking, or data interpretation issues. Although these technical issues are fundamental to interoperability, they constitute a small part of the challenges when targeting applications and services that exploit systems and data across domains, organizations, and countries. The vision of smart cities, for example, requires interoperability of IT and OT systems of different and diverse organizations, public and private, with different management frameworks and limited by several and/or differing legal constraints, including intellectual property and management of personal data. The same holds for health services, which may combine information ranging from patient monitoring to medicine management to insurance. Although most current technical efforts address fundamental issues for achieving interoperability, novel frameworks as well as architectures are required to address interoperability in a holistic way, from system to legal requirements, to achieve the vision of seamless services within or to organizations, governments, and citizens. This is crucial considering the emerging requirements for safety and security of persons, systems, and data in light of increasing risks, differing legal environments, and the dramatic emergence and deployment of machine learning and artificial intelligence at all fronts. Overall, several challenges need to be addressed and overcome to achieve interoperability and reap its benefits. They include technical barriers, security and privacy risks, semantic issues, organizational/operational incompatibilities, lack of standards and protocols, trust issues, data silos, proprietary closed systems, and legal restrictions.

Interoperability has taken a boost in the last few years because of the emerging digitalization worldwide. The COVID-19 lockdowns initiated a further deployment, although hasty at times, for services to citizens and businesses, especially for e-governance. Thus, e-government has been a leader in addressing interoperability challenges considering the urgent need for digitized citizen lifetime milestones (birth, death, retirement) and the regulation of resources, such as power, water, and medical supply chains.

FRAMEWORK FOR INTEROPERABILITY

E-government depends strongly on interoperability, which offers numerous benefits by improving communication, data sharing, and collaboration among public administrations, covering Administration to Citizen (A2C), Administration to Administration (A2A), and Administration to Business (A2B) interactions, as shown in Figure 1.[3] The following benefits result:

- *Citizen-centric services*: Citizens receive faster and more convenient services through interconnected government systems.
- *One-stop services*: The use of a centralized digital platform provides citizens and businesses access to multiple government services in one place, eliminating the need to visit different offices or websites for various services by integrating them into a single portal, simplifying interactions with the government, reducing bureaucracy, and improving efficiency.
- *Automation of processes*: This streamlines workflows, cutting down administrative costs and errors.
- *Simplified user experience*: Citizens gain easier access to

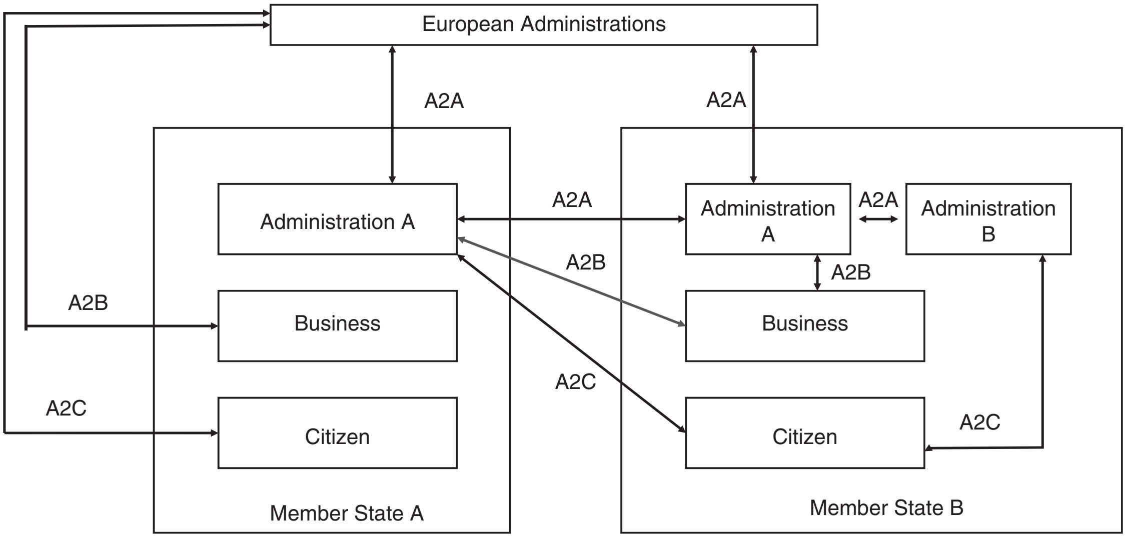


FIGURE 1. Interaction types in the context of e-government. (Source: European Commission 2004,[3] p. 13.)

services, improving satisfaction and trust in government.

- *Traceability*: Integrated systems help track and audit processes, reducing opportunities for fraud or corruption.
- *Reduced redundancy*: This avoids duplication of data and processes across agencies.
- *Resource optimization*: The shared infrastructure and services reduce operational costs.

In 2003, the European Union (EU) acknowledged the difficulty of achieving effective and efficient e-government and the necessity for information sharing among different parts and levels of government in Europe: “Full-scale implementation of eGovernment raises difficult issues. These include safeguarding trust and confidence in online interaction with governments, widespread access to online services so that no digital divide is created, interoperability for information exchange across organizational and national borders, and advancing pan-European services that support mobility in the Internal Market and European Citizenship.”[4]

Recognizing the need for interoperability among public administrations, the European Commission has progressively developed and updated the EU interoperability strategy, defined corresponding frameworks for interoperability between Member State public administrations, and funded actions to develop, promote, and deploy interoperability solutions in Member States. Specifically, the EU developed and evolved the European Interoperability Framework (EIF), a set of standards and guidelines that describes the way in which organizations have agreed, or should agree, to interact with each other. The goal of the EIF is 1) to inspire European public administrations to design and deliver seamless European public services to other public administrations, citizens, and businesses and 2) to provide guidance to Member States to design and update their national interoperability frameworks, national policies, strategies, and guidelines promoting interoperability. The latest of the three consecutive EIF versions was issued in 2017.[5]

The EIF adopts a four-layer reference model, identifying the different interoperability aspects to address when designing and delivering effective and efficient public services. The layers correspond to different, hierarchical aspects that are interdependent and required for complete and correct services: legal, organizational, semantic, and technical. Similar to layered models, higher layers require the services of lower layers. More specifically:

- *Legal interoperability* ensures that organizations operating under different legal frameworks, policies, and strategies are able to work together. This requires that legislation does not block the establishment of European public services within and between Member States. It also may require clear agreements about how to deal with differences and incompatibilities in legislation among public administrations within a Member State and across borders, including the option of putting in place new legislation.
- *Organizational interoperability* refers to the way in which public administrations align their business processes, responsibilities, and expectations to achieve commonly agreed-on and mutually beneficial goals. This requires documenting and integrating or aligning business processes and relevant information exchanged.
- *Semantic interoperability*, covering both syntactic and semantic aspects, ensures that the precise format and meaning of exchanged data and information are preserved and understood throughout exchanges between parties. In other words, ”what is sent is what is understood.”
- *Technical interoperability* includes aspects such as interface specifications, interconnection services, data integration services, data presentation and exchange, and secure communication protocols.

The EIF serves as the key reference for implementing interoperability in the European public sector. To remain relevant, it must be regularly reviewed and updated.[6]

Although the EIF has been developed for e-government, it is clearly applicable to all domains, although some may not need all layers. The EIF is increasingly used as the foundation to develop specialized interoperability frameworks, such as the eHealth EIF[a] and the EIF for Smart Cities and Communities.[b]

CHALLENGES AND RISKS

Interoperability is often viewed as merely a technical issue, but focusing solely on technical aspects is clearly not enough to efficiently connect administrations, dataflows, and services. The issue requires not only technical compatibility but also semantic clarity for data exchange and processing, while establishing the necessary organizational and legal frameworks, such as rights for data access, exchange, and reuse.

Interoperability challenges can arise at all levels of the EIF model: technical, semantic, organizational, and legal. For example, at the *technical* level, there are several known problems that relate to data quality, presenting a significant challenge that directly affects how systems can exchange, interpret, and process data effectively. Inconsistent, inaccurate, incomplete, or outdated data lead to failures in system integration and decision making. Even well-defined application programming interfaces are limited by issues of data quality.

At the *semantic* level, even high-quality data can lead to failing services caused by semantic inconsistencies. A typical example is the confusion of semantics of fields among communicating

[a] https://data.europa.eu/doi/10.2759/10138
[b] https://data.europa.eu/doi/10.2799/816559

systems, such as the differentiation between a parent and a custodian in school data management systems. We have experienced several cases where legacy school management systems, designed as isolated solutions for specific public organizations, either stored solely parent data or did not require a distinction between parent and custodian roles. Integrating these systems to larger environments that enforce distinct roles with varying data access rights can lead to interoperability problems, such as provision of an underage child's information to a nonentitled user (for example, a parent who has lost custody) or no provision of information to a user who is entitled to it (a custodian). Such role differentiations are frequent regarding systems and persons and require managerial, organizational, and legal interventions to define semantics appropriately.

The lack of a unique identifier for a system or a person challenges interoperability at the *organizational* level. This lack creates difficulties in managing and sharing data across different systems, organizations, or domains (for example, education, social services, and health). Without consistent and universal identifiers, organizations cannot link and correlate a person's or a system's information across various services or platforms, leading to inefficiencies, errors, and fragmented service delivery.

Data protection and privacy regulations can be a *legal* barrier to interoperability. For example, a government health department may need to share a citizen's health records with the social services department to determine eligibility for a welfare program. However, under the General Data Protection Regulation, this sharing of personal health data is constrained and can occur only under specific conditions. Data ownership is an issue that needs to be associated and addressed accordingly.

Privacy is a major concern in the emerging landscape of interoperable systems and services. A fragmented approach to protection of privacy, for example, at the service level, may enable providers to collect sufficient personal information through several services to build user, process, or system profiles that are questionable, at least in terms of legitimacy or ethics.

Clearly, there is still a lot of work to be done to augment frameworks like the EIF to address nonfunctional issues, such as performance, real-time requirements, and safety.

Interoperability is key to the provision of value-added services in domains ranging from the industrial floor to e-governance and policy making. Addressing the coordination and cooperation of heterogeneous systems, organizations, and even nations for data exchange and service provision, interoperability requires a coordinated and systematic treatment through frameworks, like the EIF, that include all aspects, from technical to legal. Such a holistic and systematic approach is necessary not only for effective and efficient services but for the protection and appropriate use of data and the resulting safety of persons, processes, organizations (public and private), and governments. C

REFERENCES

1. A. Tolk and J. A. Muguira, "The levels of conceptual interoperability model," in *Proc. Fall Simul. Interoperability Workshop*, 2003, pp. 1–11. Accessed: Feb. 15, 2025. [Online]. Available: https://www.researchgate.net/publication/240319008_the_levels_of_conceptual_interoperability_model
2. S. Acharya, A. A. Khan, and T. Päivärinta, "Interoperability levels and challenges of digital twins in cyber–physical systems," *J. Ind. Inf. Integration*, vol. 42, Nov. 2024, Art. no. 100714, doi: 10.1016/j.jii.2024.100714.
3. "European interoperability framework for pan-European eGovernment services." Publications Office of the EU. Accessed: Feb. 15, 2025. [Online]. Available: https://op.europa.eu/en/publication-detail/-/publication/a4778634-27fa-43b4-9912-f753c4fdfc3f
4. "Communication from the Commission to the Council, the European Parliament, the European Economic and Social Committee and the Committee of the Regions - The role of eGovernment for Europe's future [SEC(2003) 1038]." European Union. Accessed: Feb. 15, 2025. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52003DC0567
5. European Commission: Directorate-General for Digital Services. "New European interoperability framework—Promoting seamless services and data flows for European public administrations." Publications Office of the EU. Accessed: Feb. 23, 2025. [Online]. Available: https://data.europa.eu/doi/10.2799/78681
6. "Communication from the Commission to the European Parliament, The Council, The European Economic and Social Committee and the Committee of the Regions on a strengthened public sector interoperability policy linking public services, supporting public policies and delivering public benefits towards an 'Interoperable Europe'." European Union. Accessed: Feb. 21, 2025. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52022DC0710

NIKOLINA RENIERI is head of the Interoperability and Standardization Department at the Computer Technology Institute and Press "Diophantus," Patras 26504, Greece. Contact her at renieri@cti.gr.

DIMITRIOS SERPANOS is president of the Computer Technology Institute and Press "Diophantus" and a professor at the University of Patras, Patras 26504, Greece. He is a Senior Member of IEEE. Contact him at serpanos@computer.org.

EDITOR PHIL LAPLANTE
IEEE Fellow;
plaplante@psu.edu

COPYRIGHT ISTOCKPHOTO, CREDIT:LVCANDY

Citizen Development, Low-Code/No-Code Platforms, and the Evolution of Generative AI in Software Development

J. T. Sodano, EPAM Systems
Joanna F. DeFranco, The Pennsylvania State University

The demand for faster software solutions exceeds the supply of skilled software developers. More businesses will adopt citizen development frameworks and generative AI tools; however, this solution adds some challenges for project governance and security.

Digital Object Identifier 10.1109/MC.2025.3547073
Date of current version: 28 April 2025

Many organizations are supported by software—but there is a shortage of software developers and engineers, which could cause lost revenue.[1] This is leading to empowerment in building applications using low-code/no-code (LCNC) platforms enabling faster solutions. This democratization of information technology (IT) continues to accelerate, enabling broader participation in software creation beyond traditional IT departments and software engineers.[2]

This trend has led to "citizen developers," who are nontechnical users leveraging drag-and-drop tools and prebuilt components to create digital solutions for specific needs. LCNC platforms have been instrumental to this movement by providing a means for nonprofessional developers to rapidly build business applications.[3,4] Meanwhile, generative artificial intelligence (GenAI) has more recently emerged as an enabler for software development.[5] GenAI platforms, powered by large language models (LLMs), can produce or refactor code with minimal input using natural language, further reducing the barriers to entry for software innovation. However, as these tools become more intuitive and easier to use, questions arise as to whether they will introduce new risks or governance challenges, particularly in parallel with existing LCNC solutions.[5,6]

The core issue lies in determining how best to integrate the use of GenAI tools into citizen development without compromising application quality, security, and governance. LCNC platforms already address some portion of the skill gap by minimizing dependencies on complex programming languages.[7] However, the capabilities of GenAI code assistants/tools add a new layer of complexity, and the topic of how GenAI code features may complement or potentially replace traditional LCNC functionality has not yet been fully explored. In this article, we look to explore the intersecting roles of citizen development, LCNC platforms, and GenAI code systems while highlighting best practices and governance strategies that can help organizations manage the transition toward increased technology democratization if it is right for the business.

BENEFITS AND CHALLENGES OF CITIZEN DEVELOPMENT

Citizen development refers to software creation by nontechnical individuals with little to no programming skills. These are typically domain experts who lack formal training in software development and programming.[2] In many organizations, these citizen developers emerge to address gaps left by resource-constrained IT teams. Their projects often address immediate business needs such as workflow automation, data collection, and niche analytics.[3] This democratized approach can both complement and challenge the conventional enterprise IT model, where development is managed and controlled by specialized software engineers.[4]

The proliferation of citizen development has produced several tangible benefits. *Time to market* for solution innovation can be accelerated when domain experts are able to create prototypes and even entire applications more rapidly than through traditional programming scenarios, resulting in expedited digital transformation initiatives.[8] Because these citizen developers have firsthand knowledge of specific business needs, applications are often more *closely aligned with user requirements*.[9,10,11] Meanwhile, organizations often achieve *cost savings* when citizen developers reduce workloads on specialized development resources.[11,12] With increased cross-functional contribution, a heightened sense of engagement between business and IT stakeholders can *increase overall technology adoption*.[2]

Despite these advantages, some significant challenges remain. *Software quality* can vary substantially because nonexperts lack grounding in security and architecture principles.[5,10,13] *Fragmentation in governance models* also enables unmonitored "shadow IT" to grow where solutions evolve outside of sanctioned organizational oversight.[10,14] The potential for *vulnerabilities, integration issues, and application sprawl* grows absent standardized frameworks. Nevertheless, as organizations contend with market demands and IT capacity constraints, citizen development continues to increase on the basis of a growing number of tools that reduce or outright eliminate the need for coding knowledge.[9,10,11,12,13]

THE ROLE OF LCNC PLATFORMS IN CITIZEN DEVELOPMENT

LCNC platforms rely on graphical user interfaces, prebuilt modules, and configuration-driven workflows to simplify or eliminate direct source code writing.[3,4] These tools have matured significantly over time, offering features such as drag-and-drop design elements, automated database integration, and rule-based logic flows.[7] In practical terms, these features provide entry points for citizen developers by reducing the learning curve and automating a significant portion of the technical foundation typically required in traditional programming.

LCNC platforms often provide templates for common business processes and automated consistency checks. Some platforms may also integrate with advanced analytics or data visualization functions that grant users the ability to incorporate sophisticated capabilities without delving into low-level code.[10,15] While these tools may accelerate productivity, poorly governed LCNC deployments can produce redundant applications or security issues when organizations fail to coordinate efforts. Additionally, some domain experts still struggle with abstract design principles or logic flows embedded within graphical interfaces.[2] Large-scale applications introduce another level of complexity when LCNC-developed solutions must integrate with enterprise systems and adhere to the same security and performance standards as conventional software.[4] Despite these constraints, the trend toward LCNC platforms continues to grow because a structured environment where citizen developers can innovate rapidly and with minimal programming skills and fewer barriers is accessible.

THE EMERGENCE OF GENERATIVE AI CODE IN CITIZEN DEVELOPMENT

GenAI is set to redefine who can write software and how they do it, particularly in the context of citizen development. Although a low to moderate basic scripting skill level may be required, recent advances in LLMs allow AI to recommend or auto-generate entire blocks of code using natural language prompts.[5] This evolution has drawn from extensive training based on open source libraries and other code repositories, resulting in pattern-based predictions for a variety of coding tasks.[5,6]

For citizen developers, GenAI functionality can dramatically reduce complexity. If the user can state in language the desired outcome of an application, AI can propose solution logic that addresses the request.[5,15] This process supports faster prototyping and refinements, allowing people with limited coding backgrounds to iterate quickly. GenAI suggestions may also apply a framework based on best practices recognized from the AI's training corpus that decreases human errors and strengthens consistency in the final software output.[5]

Nevertheless, this approach raises a number of concerns. Even if the GenAI code syntax is correct, it *may fail to meet functional requirements* if the prompts provided are ambiguous or incomplete.[6] Within enterprise environments, questions about security and intellectual property are heightened with a *risk* that GenAI may produce code snippets that draw in part from licensed or proprietary code. Citizen developers who already struggle with verifying LCNC build solutions may find themselves even more challenged when validating machine-generated logic. In addition, the risk of unknowingly introducing malicious code or violating legal boundaries increases when users blindly accept AI outputs.[5] These issues highlight the need for further study on how GenAI can best integrate with LCNC platforms to enable organizations to benefit from faster development without compromising quality or governance.

A HYBRID MODEL INTEGRATING LCNC AND GAI TOOLS

A hybrid approach that merges the relative reliability of LCNC platforms with the versatility of GenAI models could fundamentally reshape citizen development. In this paradigm, visual workflows and structured components from LCNC systems can operate alongside real-time AI code suggestions, enabling quicker and more adaptable software development.[5,9,15] The key challenge is to integrate both in a way that respects organizational policies, safeguards security, and ensures that nontechnical contributors remain empowered rather than overwhelmed.

Comprehensive training and enablement must be part of this hybrid model. Citizen developers benefit from clear guidelines on how to craft effective prompts for AI and interpret the generated code in ways consistent with their organization's quality controls.[15] Mentorship programs that pair novices and experienced staff, or "centers of excellence," can mitigate the risks of placing too much trust in automated suggestions.[9,15] A complementary governance framework consisting of role-based access, structured reviews, and mandatory testing before production deployment can limit the potential for shadow IT scenarios.[3] In this scenario, domain experts continue to innovate while IT professionals provide oversight and ensure alignment with broader enterprise standards.

Testing and validation procedures have increased importance when combining LCNC and GenAI code capabilities. Automated tools that detect anomalies, security flaws, or accessibility issues should be run continuously as a part of the development flow.[8,16] Code reviews, assisted by separate AI modules, may be used to confirm that new logic follows best practices. In regulated industries, potential compliance enforcement tools with the ability to scan for data privacy violations could be integrated directly with LCNC platforms and GenAI engines.[5] The iterative process of automated checks followed by human validation ensures coherence across various citizen development initiatives.[16]

A WAY FORWARD

The intersection of citizen development, LCNC platforms, and GenAI represents a pivotal shift in how software is conceived, built, and governed. By extending development capabilities to a broader array of contributors, organizations can discover new paths to innovation and problem-solving during a time when competitive advantages rely on rapid digital transformation. Incorporating GenAI into these processes can further reduce barriers to entry, particularly for individuals without formal programming backgrounds.

This continuous evolution presents simultaneous challenges for project governance, application security, and the responsible use of AI-driven code suggestions. It remains uncertain how to effectively address data privacy and intellectual property concerns in this context as well as whether organizations can create standardized guidelines that balance flexibility for citizen developers with compliance with enterprise standards. To facilitate the seamless integration of GenAI into LCNC workflows, organizations should prioritize establishing best practices for oversight, implementing rigorous testing protocols, and adopting formal training programs focused on enhancing critical-thinking skills among citizen developers.

Future research must examine how best to integrate advanced AI features alongside existing LCNC functionalities. Furthermore, the organizational design implications of this hybrid development model warrant additional investigation. It is possible that new roles will emerge to serve as AI "prompt engineers," bridging the communication gap between domain experts and AI engines. Meanwhile, more sophisticated governance strategies or automated compliance mechanisms may evolve to further mitigate the challenges associated with these novel coding partnerships.

GenAI is well positioned to further the existing trends in citizen development. It has the potential to drive unprecedented innovation within enterprise software while also enhancing efficiency and empowerment for a broader and more diverse community of developers. C

REFERENCES

1. K. Madding, "Developer to accelerate business efficiency," *Forbes*, Jan. 31, 2023. [Online]. Available: https://www.forbes.com/councils/forbestechcouncil/2023/01/31/the-rise-of-the-citizen-developer-to-accelerate-business-efficiency/
2. D. Hoogsteen and H. Borgman, "Empower the workforce, empower the company? Citizen development adoption," in *Proc. 55th Hawaii Int. Conf. Syst. Sci.*, 2022, pp. 4417–4726, doi: 10.24251/HICSS.2022.575.
3. S. A. A. Naqvi, M. P. Zimmer, R. Syed, and P. Drews, "Understanding the socio-technical aspects of low-code adoption for software development," in *Proc. ECIS Res. Papers, 357*. Kristiansand, Norway: Association for Information Systems, 2023. [Online]. Available: https://aisel.aisnet.org/ecis2023_rp/357
4. M. Overeem and S. Jansen, "Proposing a framework for impact analysis for low-code development platforms," in *Proc. ACM/IEEE Int. Conf. Model Driven Eng. Lang. Syst. Companion (MODELS-C)*, 2021, pp. 88–97, doi: 10.1109/MODELS-C53483.2021.00020.
5. O. Bruhin, E. Dickhaut, E. Elshan, and M. M. Li, "The rise of generative AI in low code development platforms—An analysis and future directions," in *Proc. 57th Hawaii Int. Conf. Syst. Sci.*, 2024, pp. 7780–7789, doi: 10.24251/HICSS.2023.932.
6. S. Bubeck et al., "Sparks of artificial general intelligence: Early experiments with GPT-4," 2023, *arXiv:2303.12712*.
7. C. Silva, J. Vieira, J. C. Campos, R. Couto, and A. N. Ribeiro, "Development and validation of a descriptive cognitive model for predicting usability issues in a low-code development platform," *Human Factors*, vol. 63, no. 6, pp. 1012–1032, 2021, doi: 10.1177/0018720820920429.
8. S. Rafi, M. A. Akbar, M. Sánchez-Gordón, and R. Colomo-Palacios, "DevOps practitioners' perceptions of the low-code trend," in *Proc. ACM/IEEE Int. Symp. Emp. Softw. Eng. Meas. (ESEM)*, New York, NY, USA: ACM, 2022, pp. 1–6, doi: 10.1145/3544902.3546635.
9. B. Binzer and T. J. Winkler, "Low-coders, no-coders, and citizen developers in demand: Examining knowledge, skills, and abilities through a job market analysis," in *Proc. 18th Int. Conf. Wirtschaftsinformatik*, Nuremberg, Germany: Association for Information Systems, 2023, pp. 123–130. [Online]. Available: https://aisel.aisnet.org/wi2023/17
10. N. Callinan and M. Perry, "Critical success factors for citizen development," *Open J. Appl. Sci.*, vol. 14, no. 4, pp. 1121–1149, 2024, doi: 10.4236/ojapps.2024.144073.
11. J. Kirchhoff, N. Weidmann, S. Sauer, and G. Engels, "Situational development of low-code applications in manufacturing companies," in *Proc. ACM/IEEE 25th Int. Conf. Model Driven Eng. Lang. Syst. (MODELS) Companion*, New York, NJ, USA: ACM, 2022, pp. 1–10, doi: 10.1145/3550356.3561560.
12. B. Adrian, S. Hinrichsen, and A. Nikolenko, "App development via low-code programming as part of modern industrial engineering education," in *Advances in Human Factors and Systems Interaction*, I. L. Nunes, Ed., vol. 1207, Cham, Switzerland: Springer-Verlag, 2020, pp. 45–51.
13. E. Elshan, D. Germann, E. Dickhaut, and M. Li, "Faster, cheaper, better? Analyzing how low code development platforms drive bottom-up innovation," in *Proc. ECIS Research-in-Progress Papers*. Kristiansand, Norway: Association for Information Systems, 2023, pp. 1–10. [Online]. Available: https://aisel.aisnet.org/ecis2023_rip/82
14. M.-E. Godefroid, R. Plattfaut, and B. Niehaves, "IT outside of the IT department: Reviewing lightweight IT in times of shadow IT and IT consumerization," in *Innovation Through Information Systems*, F. Ahlemann, R. Schütte, and S. Stieglitz, Eds., vol. 48, Cham, Switzerland: Springer-Verlag, 2021, pp. 554–571.
15. V. Berardi, V. Kaur, D. Thacker, and G. Blundell, "Towards a citizen development andragogy: Low-code platforms, design thinking, and knowledge-based dynamic capabilities," *Int. J. Higher Educ. Manage.*, vol. 9, no. 2, pp. 1–21, 2023, doi: 10.24052/IJHEM/V09N02/ART-1.
16. V. S. Barletta, F. Cassano, A. Pagano, and A. Piccinno, "New perspectives for cyber security in software development: When end-user development meets artificial intelligence," in *Proc. Int. Conf. Innov. Intell. Inform. Comput. Technol. (3ICT)*. Piscataway, NJ, USA: IEEE Press, 2022, pp. 531–534, doi: 10.1109/3ICT56508.2022.9990622.

J. T. SODANO is head of Digital Workplace at EPAM Systems, a leading global provider of digital engineering, software development, and consulting services headquartered in Newtown, PA 19020 USA, and a student in the Doctor of Engineering program at The Pennsylvania State University, University Park, PA 16802 USA. Contact him at jsodano@psu.edu.

JOANNA F. DeFRANCO is an associate professor of software engineering at The Pennsylvania State University, University Park, PA 16802 USA, and an associate editor in chief of *Computer*. Contact her at jfd104@psu.edu.

IEEE COMPUTER SOCIETY

www.computer.org

PURPOSE: Engaging professionals from all areas of computing, the IEEE Computer Society sets the standard for education and engagement that fuels global technological advancement. Through conferences, publications, and programs, IEEE CS empowers, guides, and shapes the future of its members, and the greater industry, enabling new opportunities to better serve our world.

OMBUDSMAN: Contact ombudsman@computer.org.

CHAPTERS: Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

PUBLICATIONS AND ACTIVITIES

***Computer*:** The flagship publication of the IEEE Computer Society, *Computer*, publishes peer-reviewed technical content that covers all aspects of computer science, computer engineering, technology, and applications.

Periodicals: The IEEE CS publishes 12 magazines, 18 journals

Conference Proceedings & Books: Conference Publishing Services publishes more than 275 titles every year.

Standards Working Groups: More than 150 groups produce IEEE standards used throughout the world.

Technical Communities: TCs provide professional interaction in more than 30 technical areas and directly influence computer engineering conferences and publications.

Conferences/Education: The IEEE CS holds more than 215 conferences each year and sponsors many educational activities, including computing science accreditation.

Certifications: The IEEE CS offers three software developer credentials.

AVAILABLE INFORMATION

To check membership status, report an address change, or obtain information, contact help@computer.org.

IEEE COMPUTER SOCIETY OFFICES

WASHINGTON, D.C.:
2001 L St., Ste. 700,
Washington, D.C. 20036-4928
Phone: +1 202 371 0101
Fax: +1 202 728 9614
Email: help@computer.org

LOS ALAMITOS:
10662 Los Vaqueros Cir.,
Los Alamitos, CA 90720
Phone: +1 714 821 8380
Email: help@computer.org

IEEE CS EXECUTIVE STAFF

Executive Director: Melissa Russell
Director, Governance & Associate Executive Director: Anne Marie Kelly
Director, Conference Operations: Silvia Ceballos
Director, Information Technology & Services: Sumit Kacker
Director, Marketing & Sales: Michelle Tubb
Director, Membership Development: Eric Berkowitz
Director, Periodicals & Special Projects: Robin Baldwin

Digital Object Identifier 10.1109/MC.2025.3534544

IEEE

revised 14 January 2025

IEEE CS EXECUTIVE COMMITTEE

President: Hironori Washizaki
President-Elect: Grace A. Lewis
Past President: Jyotika Athavale
Vice President: Nils Aschenbruck
Secretary: Yoshiko Yasuda
Treasurer: Darren Galpin
VP, Member & Geographic Activities: Andrew Seely
VP, Professional & Educational Activities: Cyril Onwubiko
VP, Publications: Charles (Chuck) Hansen
VP, Standards Activities: Edward Au
VP, Technical & Conference Activities: Terry Benzel
2025–2026 IEEE Division VIII Director: Cecilia Metra
2024–2025 IEEE Division V Director: Christina M. Schober
2025 IEEE Division V Director-Elect: Leila De Floriani

IEEE CS BOARD OF GOVERNORS

Term Expiring 2025:
İlkay Altintaş, Mike Hinchey, Joaquim Jorge, Rick Kazman, Carolyn McGregor, Andrew Seely

Term Expiring 2026:
Megha Ben, Terry Benzel, Mrinal Karvir, Andreas Reinhardt, Deborah Silver, Yoshiko Yasuda

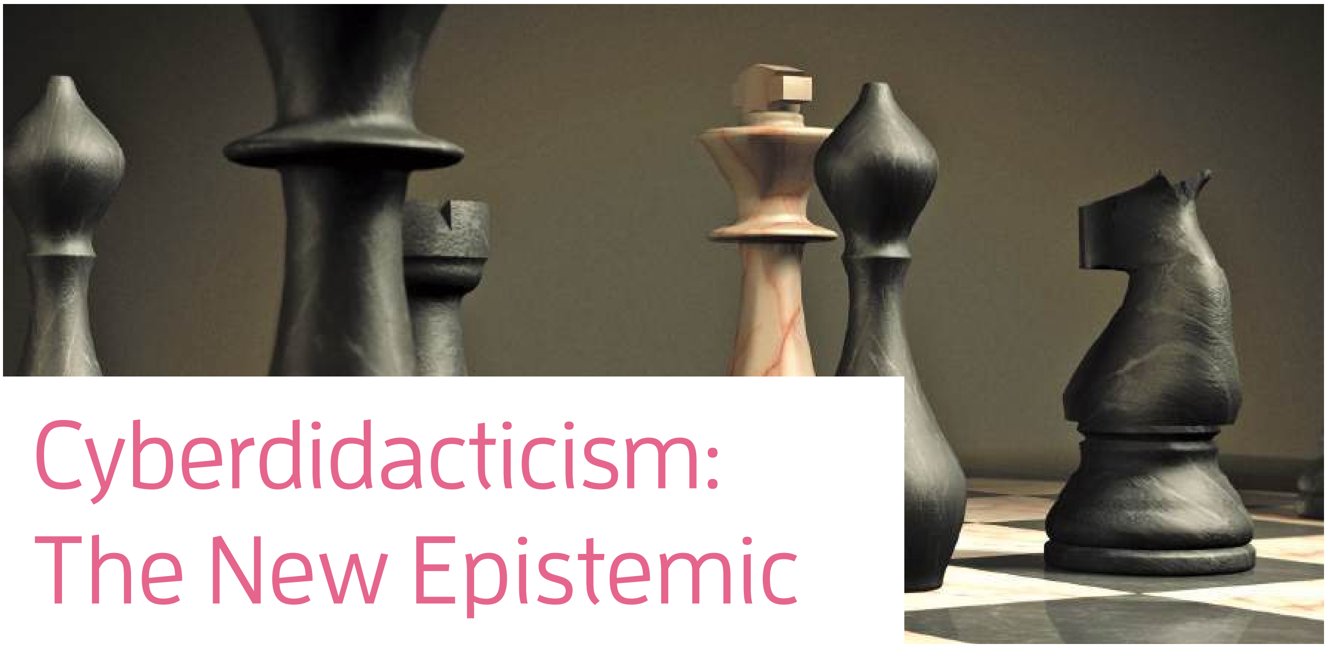
Term Expiring 2027:
Sven Dickinson, Alfredo Goldman, Daniel S. Katz, Yuhong Liu, Ladan Tahvildari, Damla Turgut

IEEE EXECUTIVE STAFF

Executive Director and COO: Sophia Muirhead
General Counsel and Chief Compliance Officer: Ahsaki Benion
Chief Human Resources Officer: Cheri N. Collins Wideman
Managing Director, IEEE-USA: Russell Harrison
Chief Marketing Officer: Karen L. Hawkins
Managing Director, Publications: Steven Heffner
Staff Executive, Corporate Activities: Donna Hourican
Managing Director, Member and Geographic Activities: Cecelia Jankowski
Chief of Staff to the Executive Director: Kelly Lorne
Managing Director, Educational Activities: Jamie Moesch
IEEE Standards Association Managing Director: Alpesh Shah
Chief Financial Officer: Thomas Siegert
Chief Information Digital Officer: Jeff Strohschein
Managing Director, Conferences, Events, and Experiences: Marie Hunter
Managing Director, Technical Activities: Mojdeh Bahar

IEEE OFFICERS

President & CEO: Kathleen A. Kramer
President-Elect: Mary Ellen Randall
Past President: Thomas M. Coughlin
Director & Secretary: Forrest D. Wright
Director & Treasurer: Gerardo Barbosa
Director & VP, Publication Services & Products: W. Clem Karl
Director & VP, Educational Activities: Timothy P. Kurzweg
Director & VP, Membership and Geographic Activities: Antonio Luque
Director & President, Standards Association: Gary R. Hoffman
Director & VP, Technical Activities: Dalma Novak
Director & President, IEEE-USA: Timothy T. Lee



Cyberdidacticism: The New Epistemic Paradigm for Cognitive Minimalism and Generative Artificial Intelligence

Hal Berghel, University of Nevada, Las Vegas

We postulate that generative artificial intelligence provides a new paradigm for disruptive technologies by enabling a community of cyberdidacts.

Digital Object Identifier 10.1109/MC.2025.3548920
Date of current version: 28 April 2025

Educators have always been fascinated by, and enamored of, autodidacts. There's just something inherently uplifting about individuals who can master subjects on their own. For bright, passionate, self-motivated students driven by insatiable curiosity, autodidacticism is an ideal complement to formal education. It might be an adequate replacement for traditional education in such cases were it not for the fact that its viability is highly dependent on so many external factors: environment, social circumstance, access to resources, opportunities, individual personality, genetics, etc. Further, a pseudoautodidacticism in the hands of the parochial and illiberal can quickly be driven off the rails by ideological biases and prejudice. So, while autodidacticism may not be an optimal learning environment for many, if not most, students, it is optimal for some students and refreshing for a teacher to witness.

With autodidacticism, a teacher is primarily a facilitator—someone

who identifies and provides access to resources, identifies alternative educational pathways, makes recommendations based on experience, and, above all, avoids impeding the student's progress. In this sense, the teacher is somewhat akin to a crew coxswain: useful for direction but accounting for little of the expended effort.

LEARNING AND PERSONALITY

Psychological models of human personality identify at least a half dozen or so primary traits within human personality inventories. The Big Five[1,2] and Revised NEO[3] models list some variation of these five traits: conscientiousness, agreeableness, extroversion/introversion, openness to experience, and emotional stability, while other models add to this list (for example, the HEXACO model adds a sixth: honesty–humility[4]). Psychologists have been exploring the relationship between personality traits and other human characteristics for some time. Of particular interest here is the relationship between personality traits and personal values[5] and between personality traits and academic performance.[6]

Albert Bandura's notion of self-efficacy is a pivotal concept in this regard.[7] Bandura considers self-efficacy to be an individual's confidence in his/her ability to successfully complete a task. We note that self-efficacy is a perception or feeling that is experientially acquired by individuals and thus is both positively and negatively reinforced by actual successes and failures. Self-efficacy is both transferable to similar situations and also generalizable to new situations that are different from those already experienced. Self-efficacy may also be vicarious based on the observation of others. On Bandura's account, over time, an increase (decrease) of self-efficacy produces a confidence (apprehension) when faced with new challenges. But, as Bandura cautions, "analysis of how perceived self-efficacy influences performance is not meant to imply that expectation is the sole determinant of behavior. *Expectation alone will not produce desired performance if the component capabilities are lacking* (italics added)."[7] Hold that thought. We will return to this topic later, when we show how harmful it is when inflated self-efficacy becomes a surrogate for critical capabilities, such as reasoning proficiency, knowledge, and understanding, leading to a deluded self-efficacy.

SELF-EFFICACY AND INTERACTION

Bandura's explanation that self-efficacy is a function of "experienced mastery" places it squarely within the scope of informatics[8]—the discipline that Robin Milner calls the science of interactive systems and that many consider to be the nexus of technology, domain knowledge, and people.[9] That is, the process by means of which self-efficacy is achieved, the experienced mastery if you will, circumscribes a general-purpose, interactive learning system with multisourced and many-directional information stimuli, memory, a cognitive framework, feedback mechanisms, recognizers and analyzers of verbal and nonverbal patterns, and so forth. This is what Milner calls ”conceptual armoury." There is an analogy between the acquisition of self-efficacy and what computer scientists call interactivity.[10] We may draw parallels between psychology and computer science descriptors as in such pairings as individuals/objects, stimuli/input, response/output, thoughts/processes, behavior/outcome, and so forth, as functionally similar pairs of elements that comprise complex systems that process and react to symbolic information in different domains. There is also a parallel between what psychologists call observational learning and what computer scientists call interactive computing. And strong cases can be made that both are nonalgorithmic since they may involve external, dynamic, interactive, or reactive events that take place concurrently with, but independent of, any ongoing processing.[10,11] Interactivity worthy of the name must accommodate inherently unpredictable responses to unanticipated external stimuli that is governed by possibly incomprehensible (at least, at the time) external influences. Letting a toddler play with a cell phone or mobile device or letting a blindfolded child drive a car are two primitive illustrations of the potentially unpredictable, nonalgorithmic nature of interactivity. Interactivity is a property of a truly open system. Human cognition is such a system: constrained in some ways, goal-directed and motivated in others, but nonetheless always open to new and unforeseen cognitive threads.

EFFICACY AND OUTCOME EXPECTANCY

Bandura draws an important distinction between outcome expectancy and efficacy expectation.

> "An outcome expectancy is defined as a person's estimate that a given behavior will lead to certain outcomes. An efficacy expectation is the conviction that one can successfully execute the behavior required to produce the outcomes. Outcome and efficacy expectations are differentiated, because individuals can believe that a particular course of action will produce certain outcomes, but if they entertain serious doubts about whether they can perform the necessary activities such information does not influence their behavior."[7]

This difference is subtle but critical to the hypothesis we will soon advance. Note that an irrational inflation of efficacy expectation may have undesirable social consequences, perhaps by overconfident bridge designers (for example, the designers of the Tacoma Narrows Bridge), the construction of poorly thought-through irrigation canals (resulting in the Salton Sea), the circumvention of U.S. Food and Drug Administration guidelines in the use of dangerous pharmaceuticals (for example, thalidomide), the failure to anticipate that some metals can rust and may not withstand heat (for example, those used in Takata airbags), that blowout preventers may not work well under high pressure (as in the BP Deepwater Horizon oil spill), the failure to admit that saying that a medical technology will work won't make it so (as in the case of Theranos), and so forth. Examples such as these led me to propose Gresham's twist on Moore's law: the world's capacity to create absurd technology doubles every 18 months.[12]

I'm endorsing what I consider to be a modest and uncontroversial claim: unjustifiably high efficacy expectations can have dangerous social consequences and justify continued vigilance. Further, the potential for danger is proportional to the lack of justification. For the sake of simplicity, and given that we're not conducting a research study in the social sciences, we may place my endorsement into more familiar, if pedestrian, terms: delusional overconfidence is undesirable and should be avoided. In fact, a healthy skepticism is always warranted— especially when it comes to technology.[13] Further, any technology that facilitates or encourages delusional overconfidence is prima facie objectionable, and its use should be discouraged.

CYBERDIDACTICISM

I'm suggesting that unbridled overconfidence is likely undesirable and shouldn't be encouraged without strong reservation. The widespread popularity of the "fake it 'til you make it" and "move fast and break things" aphorisms has to be taken with a large grain of salt: they have limited utility and, as time has shown, are all too often coincident with negative externalities. These aphorisms are serviceable components of a ″feel good″ approach to management: while they may upload the spirit and make the participants feel good about themselves and their activities, their vagueness is quickly seen to hide intellectual confusion or camouflage a technological immaturity.

From my perspective as an educator, there is substantial anecdotal evidence that generative artificial intelligence (AI) falls within the scope of these aphorisms. In terms of the preceding discussion, it unjustifiably inflates the "efficacy expectations" of typical users. This anecdotal evidence derives in part from the observed disparity between generative AI-produced homework and programming assignments on the one hand and exam scores and student interviews on the other—a level of disparity that was not observed to the same degree before generative AI use became commonplace in higher education. Of course, an anecdotal correlation is by no means proof of causation, but it does suggest a worthy topic for further study by social scientists. My intuition as a teacher tells me that a study somewhat analogous to the work of Bandura will reveal a strong connection between the reliance on generative AI and sundry behavioral affectations, such as inflated efficacy expectations, unjustified self-confidence, overreliance on the volume of output, suboptimal decision making, etc. That said, it is my intention here to explain the basis for my intuition as an educational observer and not a social scientist. I've observed the emergence of a new class of student, the cyberdidact, which for all intents and purposes may be considered an antithesis of the time-honored autodidact. It may be useful to draw some comparisons between the two.

Autodidacts derive considerable satisfaction from an ability to solve problems, achieve understanding, acquire mastery, etc., by themselves. Not in isolation, mind you, for inspiration is drawn from a variety of their own experiences, but without any formal instruction, motivation, or direction by others. To be sure, such self-learning is not without risk and not to be recommended for everyone. But when it works, autodidactism can avoid inefficiencies and distractions in traditional, compulsory mass education and may lead to remarkable results.

By contrast, a cyberdidact has a consumer-based, transactional approach to learning and problem solving and only a casual, incurious interest in understanding and mastery. On the cyberdidact's account, there is nothing particularly satisfying in the personal quest for knowledge but only in the apparent production of serviceable output. Indeed, that is the allure of generative AI: it provides an epiphanic-like endorphin rush with minimal cognitive investment. In this way, it is akin to interactive video games—but with the additional advantage of requiring less continuous interaction in order to achieve satisfying results. Armed with queries like ″how many Rs are in strawberry?″[14] or instructions like ″write a Python program to find prime factors for a set of integers," the cyberdidact's cognitive investment is complete—irrespective, mind you, of whether he/she fully comprehends the significance of the queries. To illustrate, what do the "strawberry" query and response tell the user–typist about the role of tokenization in large language models, the discordance between phonology and orthography, or the difference between orthography and semantics? How much of an understanding about number theory and factorization is required to create the program directive? In traditional intelligence, curiosity is the starting point of a creative process. With generative AI, curiosity is the end of the process.

TYPUS ERGO SCIO

An infatuation with generative AI lies in the superficial appeal of the end product embellished by the most cherished companions of a cognitive miser: intellectual economy and immediate gratification. But this intellectual parsimoniousness comes at a price. By deferring the majority of the cognitive heavy lifting to the generative AI tool, the user skirts the most fundamental components of metacognition: introspection, contextualization, reflection, reasoning, and the like. The ancient Greeks would describe generative AI as *nous*-less. What is more, this *nous-lessness* provides a fertile breeding ground for the propagation of cognitive biases, selective perception, cognitive dissonance, conspiracy theories, fake news, alternative facts, and sundry other pitfalls of inattentive and unprepared minds. The delusion behind the use of generative AI may be expressed by this corruption of Descartes' dictum: typus ergo scio (I type therefore I understand). With many audiences, the appeal of generative AI at this point seems to be presentation and optics over understanding and substance. Generative AI is more of a digital dilettante than an online oracle.

Because reasoning involves more than information retrieval, pattern recognition, and reaction, cognitive frugality carries with it a heavy cost. It understates the critical relationship of consciousness, understanding, and formal and informal logic to cognition, and it completely ignores the roles of self-correction, self-analysis, and self-criticism. A first principle of cognition is recognizing the substance and significance of an event. This requires more of us than the ability to produce an executable query. In very narrowly focused applications where such considerations are ancillary, such as may arise in automated theorem proving, calculation, pattern recognition, information retrieval, etc., generative AI is likely to be of considerable assistance to a scholar. But it is no substitute for human cognition: it may help in performing calculation, but it remains silent on why a calculation is important in the first place.

THE CYBERDIDACTIC HYPOTHESIS AND THE ONLINE DOPPELGÄNGER THOUGHT EXPERIMENT

We suggest the following hypothesis in light of our observations.

> The Cyberdidact Hypothesis: To the extent that it makes sense to correlate personality traits with academic performance, academic performance will not correlate with frequent use of, or reliance on, generative AI.

Potential corollaries: 1) those personality traits that correlate positively with cyberdidacticism are likely to correlate negatively with autodidacticism, vice versa; 2) the appeal of generative AI is inversely related to erudition; 3) generative AI is likely to lead to an unjustified, elevated self-efficacy; and 4) generative AI as a learning tool is demonstrably suboptimal. Why might this be?

We begin with Bandura's cautionary observation that "Expectation alone will not produce desired performance if the component capabilities are lacking." Self-efficacy is not a sufficient condition for academic or scholarly ability. Self-deception may also be at work. Self-efficacy is conditioned by internal and external feedback. Were one to see that certain patterns of behavior continually return high marks on exams, positive recognition from knowledgeable, respected peers, continued success in the exercise of skills, etc., one might legitimately assume some degree of self-efficacy. But, can we imagine a situation where the continuous feedback might be misleading?

Indeed, we can. Consider the case of a Loyal Online Doppelgänger— a loyal, reliable, expert online surrogate who can be counted on to take exams for you, interact with peers on your behalf, and perform your job—all via online communication systems where identity is electronically spoofed. Assume that the feedback on the doppelgänger's performance evaluations (in your name, of course) is consistently positive. But only you know of the existence of the doppelgänger, who, by assumption, will never disclose the ruse. Over time, how would the consistent, positive assessment of the doppelgänger's performance effect your self-efficacy? Remember that self-efficacy is conditioned by both internal and external feedback, but in this case all of the external feedback about your (the doppelgänger's) performance is strongly positive, but misdirected. My suggestion—which is confirmable or refutable by studies conducted by social scientists—is that self-delusion is an inevitable consequence and that, over time, a person's self-efficacy will unjustifiably increase despite the ruse and that this false sense of accomplishment will lead an individual to overconfidence, which will, in turn, lead that person to take on challenges for which he/she is underqualified. Our hypothesis predicts that a provable connection between our Loyal Online Doppelgänger thought experiment and the actual use of generative AI is obvious.

I further buttress my hypothesis by reference to the "Big 5 Model" (also known as the OCEAN model) of personality traits of basic psychology. For the present purposes, we'll use the definitions found on an online resource provided by the Harvard Graduate School of Education because it allows the online user to drill into arbitrary levels of detail and provides key references.[15]

1. *Conscientiousness*: The tendency to be organized, responsible, and hardworking
2. *Agreeableness*: The tendency to act in a cooperative, unselfish manner
3. *Neuroticism*: Emotional stability is predictability and consistency in emotional reactions, with absence of rapid mood changes. Neuroticism is a chronic level of emotional instability and proneness to psychological distress
4. *Openness to experience*: The tendency to be open to new

aesthetic, cultural, or intellectual experiences

5. *Extraversion*: An orientation of one's interests and energies toward the outer world of people and things rather than the inner world of subjective experience; characterized by positive affect and sociability.

Caveats are called for. First, models of personality types are instruments of social science, not computer science; so, my analysis represents an oversimplified discussion of the topic. Second, there is no universal agreement on which personality traits belong in the Big Five and what precise definitions should be used to describe them. Third, there is nothing that compels us to use the number 5—social scientists have used as few as two and as many as 20 traits.[16] Fourth, there are several different approaches to identifying relevant personality traits. I am neither a social scientist nor an expert on personality theory, but since I am advancing a hypothesis and not a proof, some brevity and occasional appeal to hand waving should be tolerable.

Social science research on the relationship between personality traits and self-efficacy has been conducted. In particular, the predictive powers of the Big 5 and self-efficacy on academic performance are well documented. The following paragraphs report the observed relationship between the Big 5 personality traits on the one hand and academic performance and self-efficacy on the other.[6]

> "Research shows that the Big Five traits relate to academic performance. Conscientiousness, that is, self-discipline, facilitates schoolwork by imparting preparedness. Openness, that is, imagination, helps with new modes of studying. Agreeableness, that is, compliance, increases consistency of class attendance. Extraversion, that is, sociability, hampers students' focus, and neuroticism, that is, emotional instability, is associated with test anxiety, where both traits hinder performance. Empirical support for the predictiveness of some traits is stronger than for others. For instance, 'Conscientiousness is the most robust predictor of academic performance with an average correlation of .20. '"
>
> "Self-efficacy is correlated with academic performance …. A recent meta-analysis examined 50 antecedents of academic performance and found that self-efficacy had the strongest correlation ($r = 0.59$) …. In the same study, of the Big Five traits, only conscientiousness significantly correlated with performance ($r = 0.19$). In another synthesis, which examined 105 predictors, self-efficacy was the second (after peer assessment) strongest predictor of academic achievement…."

My hypothesis derives from my strong suspicion that social science research will show an inverse correlation between some of these Big 5 traits and eagerness to rely on generative AI for academic and scholarly pursuits. I challenge the readers to consider for themselves the degree to which personality traits such as conscientiousness, self-discipline, imagination, consistency of class attendance, etc., would correlate with the reliance of generative AI for scholarly insight. Similarly, one might consider whether *unjustifiably* high self-efficacy is likely to lead to quality academic or scholarly work. I can see how it might lead to increased productivity (via automation), but productivity in isolation is not a reliable indicator of the accuracy, value, or impact of scholarship. Particularly worrisome is the reliance on generative AI for the creation of programming source code—especially when used in critical systems. In fact, one would expect that more reliable contributors to quality academic or scholarly work might be a climate of self-doubt, skepticism, agnosticism, and aporia.

That said, at this point, our hypothesis should be understood within the framework of technology education rather than social science research. From what I can tell, most postsecondary educators with whom I work agree that this hypothesis is consistent with observation in the classroom. However, social science research places much higher demands on hypothesis validation than observation and anecdotage. It remains to be seen whether this hypothesis will receive validation in that realm.

From a computing perspective, generative AI is algorithmic; thinking is not so limited. There is a dimension of human thought that is inherently nonlinear, dynamic, and interactive. Peter Wegner makes the point that interactive computation is nonalgorithmic convincingly in several articles,[10,17] and one key element of his argument is that algorithms cannot process disparate input information that was not anticipated in its design. In Wegner's words[10]:

> "The radical notion that interactive systems are more powerful problem-solving engines than algorithms is the basis for a new paradigm for computing technology built around the unifying concept of interaction…. The paradigm shift from algorithms to interaction is a consequence of converging changes in system architecture, software engineering, and human-computer interface technology…."
>
> What is more,
>
> "The irreducibility of interaction to algorithms enhances the intellectual legitimacy of computer science as a discipline distinct from mathematics and, by clarifying the nature of empirical models of computation, provides a technical rationale for calling computer science a science."

For additional details, the reader is encouraged to read Goldin, Smolka and Wegner.[11]

Wegner's argument implies that generative AI platforms, as algorithmic implementations of large language model neural nets, will never achieve parity with human thought. Such being the case, the use of generative AI algorithms can never prove to be an adequate substitute for human understanding.

CYBERDILETTANTISM

Again, our experience suggests that cyberdidacticism will hold out special appeal for cognitive misers characterized by lower academic standards, limited scholarly ability, unjustified overconfidence, indolence, etc. I emphasize once again that this does not imply that generative AI is without scholarly utility. Certainly, its use to jog memory, maximize information uptake, detect plagiarisms and forgeries, check facts, search databases, and review, debug, and document program code, and its aid in parsing, detecting plagiarism and copyright violations and authorship patterns, image recognition, language translation, modeling, address learning challenges, etc., are widely acknowledged. And if its use were restricted to such a support role in traditional learning environments, the potential downsides would be much shallower. However, when it is used as a surrogate for imagination, creativity, understanding, reasoning, etc., to create content, its overall social value comes into question. It is unfortunate that a large part of the appeal of generative AI in higher education seems to be that it provides a path of least resistance in the quest for measurable output and meeting deadlines. As such, it is a natural complement to social media for those who prefer presentation to substance, opinion to fact, belief over certainty, and approximation over accuracy, and are content to work with derivative and questionable content and to resolve problems with a minimum of critical reflection.

If left unchecked, generative AI cannot help but facilitate *cyberdilettantism* for those who are so inclined. If the goal is simply to generate plausible, token output, there is little incentive to go beyond a superficial understanding of a topic. It is the nature of the beast. Generative AI output justifies at best a participation trophy for the user who's minimally involved in the game.

A similar point was made in a recent article in the *Chronicle of Higher Education*:

> "Shriram Krishnamurthi, a computer science professor at Brown University, has noticed that as more high schools teach programming with wildly varying degrees of rigor, incoming students are increasingly showing up thinking they know more than they do. 'There's this weird thing where they are very competent at patching together some things and producing graphs that look nice,' Krishnamurthi said, 'but their understanding of what they did is pretty low.' (He added that he wasn't casting judgment on the individual winners at NeurIPS. 'There has always been and will always be a sliver of students that are extraordinarily capable,' he acknowledged. Outside of NeurIPS, high schoolers can pay companies a handsome fee[18] to coauthor academic papers, a cottage industry that's widely criticized."[19]

Of course, so-called paper mills have marketed bogus scholarship online for decades. This service is not limited to students. In a recent article in *Science*, Jeffrey Brainard reported that even "journals are awash in a rising tide of scientific manuscripts from paper mills - secretive businesses that allow researchers to pad their publication records by paying for fake papers or undeserved authorship."[20] Generative AI is becoming integral to the paper mill supply chain—by either allowing users to bypass the paper mill or allowing the paper mills to become more efficient. In either case, academic standards are undermined. In addition, the generative AI "paper mill" can create the illusion that the user has actually accomplished something. But, in the case of the "paper mill," there is no delusion about authorship. The purchaser knows full well that he/she has no cognitive investment in the effort. However, generative AI enables self-delusion, for the actual "author" is a computer, the paper is presumed unique, the process is anonymous, and there is no financial transaction recorded to betray the deception. Generative AI can be a form of scholarly chicanery on a desktop. Anyone with a computer and an Internet connection can become an immediate cyberdilettante.

THE ERA OF THE CYBERSAVANT

Generative AI provides access to computing power that usually isn't available to the general population. That would be a social good were it not for the fact that generative AI's appeal lies in the ability to use these platforms with

1. negligible cognitive investment
2. low or negative cognitive inertia
3. logical detachment from the underlying issues
4. a propensity for propagating bias and promoting agendas
5. a proclivity for disinformation with the potential consequence of producing an unjustified self-efficacy.

Therein lies the proverbial rub. Social scientists have studied the effect of inflated self-efficacy and overconfidence,[6] but they have not fully embraced the potential adverse effect of generative AI in the mix. We only partially understand the social effects of such technology-inspired self-delusion.[21]

Further, an overreliance on generative AI is but one of a number of current unhealthy trends in education. Its effects must be understood in the context of a broad decline in reading, a decline in foreign language programs, and the fact that scholarly materials are becoming less appealing to a general audience.[22] Humanities, liberal arts, and a diversified, well-rounded education have always been threatening to illiberal autocrats, dictators, and demagogues who focus on the development of compliant subjects and obedient workforces rather than a community of free thinkers who continuously challenge the existing order. So, selectively trained generative AI is a demagogue's dream. If our hypothesis is correct, the use of generative AI as a substitute for traditional scholarship is going to exacerbate many of our social–political ills. While society enjoys a very long history of deploying technology before fully understanding the negative externalities of its use, generative AI is unique in its ubiquity, ease of use, political implications, and potential for social disruption.

REFERENCES

1. L. Goldberg, "An alternative "description of personality": The big-five factor structure," *J. Personality Social Psychol.*, vol. 59, no. 6, pp. 1216–1229, 1990, doi: 10.1037/0022-3514.59.6.1216.
2. L. Goldberg, "The development of markers for the big five factor structure," *Psychological Assessment*, vol. 4, no. 1, pp. 26–42, 1992, doi: 10.1037//1040-3590.4.1.26.
3. R. McCrae, P. Costa Jr., and T. Martin, "The NEO–PI–3: A more readable revised NEO personality inventory," *J. Personality Assessment*, vol. 84, no. 3, pp. 261–270, 2005, doi: 10.1207/s15327752jpa8403_05.
4. M. Ashton et al., "A six-factor structure of personality-descriptive adjectives: Solutions from psycholexical studies in seven languages," *J. Personality Social Psychol.*, vol. 86, no. 2, pp. 356–366, 2004, doi: 10.1037/0022-3514.86.2.356.
5. S. Roccas, L. Sagiv, S. Schwartz, and A. Knafo, "The big five personality factors and personal values," *Personality Social Psychol. Bull.*, vol. 28, no. 6, pp. 789–801, 2002, doi: 10.1177/0146167202289008.
6. A. Stajkovic, A. Bandura, E. Locke, D. Lee, and K. Sergent, "Test of three conceptual models of influence of the big five personality traits and self-efficacy on academic performance: A meta-analytic path-analysis," *Personality Individual Differences*, vol. 120, pp. 238–245, Jan. 2018.
7. A. Bandura, "Self-efficacy: Toward a unifying theory of behavioral change," *Psychological Rev.*, vol. 84, no. 2, pp. 191–215, 1977, doi: 10.1037//0033-295X.84.2.191.
8. R. Milner, "Turing, computing and communication," in *Interactive Computation, The New Paradigm*, D. Goldin, S. A. Smolka, and P. Wegner, Eds. Berlin, Germany: Springer-Verlag, 2006, pp. 1–8.
9. H. Berghel, "Disinformatics: The discipline behind grand deceptions," *Computer*, vol. 51, no. 1, pp. 89–93, 2018, doi: 10.1109/MC.2018.1151023.
10. P. Wegner, "Why interaction is more powerful than algorithms," *Commun. ACM*, vol. 40, no. 5, pp. 80–91, 1997, doi: 10.1145/253769.253801.
11. D. Goldin, S. Smolka, P. Wegner, Eds., *Interactive Computation: The New Paradigm*. New York, NY, USA: Springer-Verlag, 2006.
12. H. Berghel, "Technology abuse and the velocity of innovation," *Cutter IT J.*, vol. 28, no. 7, pp. 12–17, 2015.
13. J. L. King, H. Berghel, P. G. Armour, and R. N. Charette, "Healthy skepticism," *Computer*, vol. 57, no. 11, pp. 86–91, Nov. 2024, doi: 10.1109/MC.2024.3422709.
14. A. Silberling. "Why AI can't spell 'strawberry'." TechCrunch. Accessed: Feb. 20, 2025. [Online]. Available: https://techcrunch.com/2024/08/27/why-ai-cant-spell-strawberry/
15. "The big five personality traits." Explore SEL. Accessed: Feb. 20, 2025. [Online]. Available: http://exploresel.gse.harvard.edu/frameworks/7
16. O. John, L. Naumann, and C. Soto, "Paradigm shift to the integrative Big Five taxonomy: History, measurement, and conceptual issues," in *Handbook of Personality: Theory and Research*, O. John, R. Robins, and L. Pervin Eds., New York, NY, USA: Guilford Press, 2008, pp. 114–158.
17. P. Wegner, "Interactive foundations of computing," *Theor. Comput. Sci.*, vol. 192, no. 2, pp. 315–351, 1998, doi: 10.1016/S0304-3975(97)00154-0.
18. D. Golden, and K. Purohit, "The newest college admissions ploy: Paying to make your teen a 'peer-reviewed,'" *ProPublica*, May 18, 2023. [Online]. Available: https://www.propublica.org/article/college-high-school-research-peer-review-publications
19. S. Lee, "Teens are doing AI research now. Is that a good thing?" *The Chronicle of Higher Education*, Jan. 14, 2025. [Online]. Available: https://www.chronicle.com/article/teens-are-doing-ai-research-now-is-that-a-good-thing?utm_source=Iterable&utm_medium=email&utm_campaign=campaign_12306631_nl_Academe-Today_date_20250115
20. J. Brainard, "Fake scientific papers are alarmingly common," *Science*, May 9, 2023. [Online]. Available: https://www.science.org/content/article/fake-scientific-papers-are-alarmingly-common
21. H. Berghel, "Generative artificial intelligence, semantic entropy, and the big sort," *Computer*, vol. 57, no. 1, pp. 130–135, 2024, doi: 10.1109/MC.2023.3331594.
22. S. Carlson, and N. Laff, "The hidden utility of the liberal arts," *The Chronicle of Higher Education*, Jan. 21, 2025. [Online]. Available: https://www.chronicle.com/article/the-hidden-utility-of-the-liberal-arts?utm_source=Iterable&utm_medium=email&utm_campaign=campaign_12412980_nl_Academe-Today_date_20250127

HAL BERGHEL is a professor of computer science at the University of Nevada, Las Vegas, Las Vegas, NV 89154 USA. Contact him at hlb@computer.org.

CALL FOR SPECIAL ISSUE PROPOSALS

Computer solicits special issue proposals from leaders and experts within a broad range of computing communities. Proposed themes/issues should address important and timely topics that will be of broad interest to *Computer*'s readership. Special issues are an essential feature of *Computer*, as they deliver compelling research insights and perspectives on new and established technologies and computing strategies.

Please send us your high-quality proposals for the 2025–2026 editorial calendar. Of particular interest are proposals centered on:

- 3D printing
- Robotics
- LLMs
- AI safety
- Dis/Misinformation
- Legacy software
- Microelectronics

Proposal guidelines are available at:

www.computer.org/csdl/magazine/co/write-for-us/15911

IEEE COMPUTER SOCIETY

IEEE

Digital Object Identifier 10.1109/MC.2025.3553359

CS CONNECTION

NORTH AMERICA STUDENT CHALLENGE COMPETITION: ENGAGING STUDENTS TO SOLVE REAL-WORLD DATA PROBLEMS

The IEEE Computer Society's North America Student Challenge (NASC) Competition was held between October and December of 2024. Of the 43 registered teams, three were crowned winners at the IEEE Big Data Conference in December. The teams were composed of up to three students, with an optional faculty mentor, and they were challenged to solve three problems based on real-world datasets.

The first challenge problem was predicting missing resource usage data from data center traces. The second problem was inferring latent user preference from conversations with a large language model. The final problem was predicting the invocation rate of functions in a cloud computing platform.

The first round of submissions gave teams roughly three weeks to submit their solutions. Finalists were identified based on these submissions, and the top three teams were able to attend the IEEE Big Data Conference in Washington, DC, and present in front of the panel of judges.

The judges included Deborah Silver from Rutgers University, Haoliang Wang from Adobe Research, and Kaiqun Fu from South Dakota State University. After the teams were evaluated carefully, based on their oral presentation, quantitative performance, and their solutions' novelty, the winners were ranked and awarded on 17 December 2024 at the banquet held at the Smithsonian National Museum of the American Indian.

The winning team was composed of Bilal Saleem, Syed Hasan Amin Mahmood, and Omar Basit, from Purdue University (Figure 1). The first runner-up team was made up of Daniel Leeds, Harrison Huang, and Jonathan Mak, from Rice University, and the second runner-up team was a team of one, Eliot Hall, from San José State University.

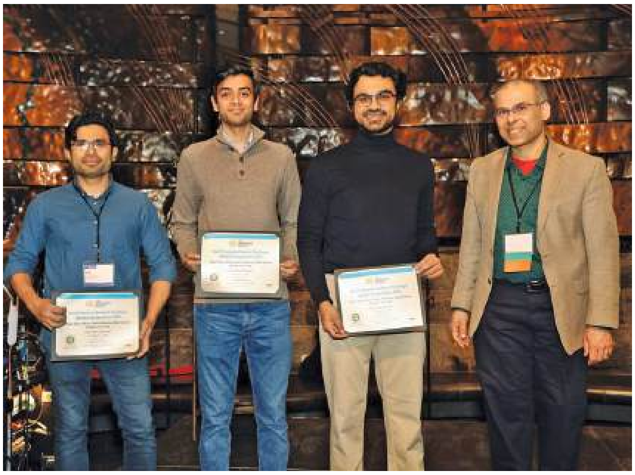


FIGURE 1. The winning team: Bilal Saleem, Omar Basit, and Syed Hasan Amin Mahmood from Purdue University, with Prof. Saurabh Bagchi, lead organizer of the NASC Competition.

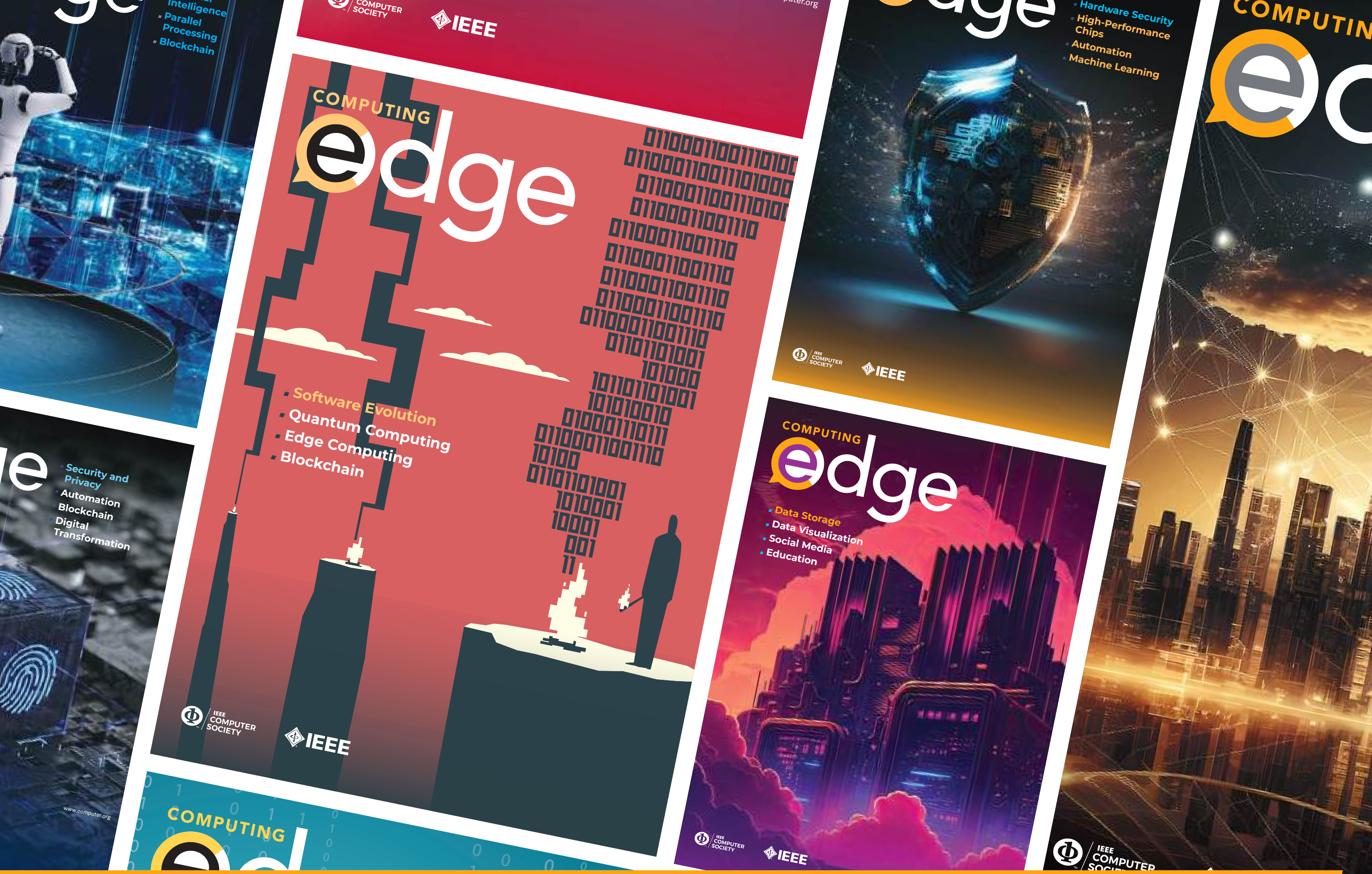
The NASC competition was an invaluable experience for many. Saurabh Bagchi, professor at Purdue University and lead organizer of the event, said that "This kind of event energizes our student community and makes the participants realize that they are part of a community rather than isolated islands. This also brings together industry practitioners and academic researchers to formulate meaningful challenge problems."

The event was supported by the IEEE Computer Society (CS) Board of Governors and cosponsored by Adobe. Fellow IEEE CS Board members, Joaquim Jorge (University of Lisbon) and Deborah Silver (Rutgers University), joined Bagchi in organizing the event from proposal to execution. In light of the event's success, the team is planning on expanding the geographical scope in 2025, and running it as the IEEE CS Global Student Challenge (GSC) Competition.

For more information on the 2024 NASC Competition and for a recording of the final presentations, please visit the CS website: https://www.computer.org/publications/tech-news/events/north-america-student-challenge-2024.

Digital Object Identifier 10.1109/MC.2025.3539809
Date of current version: 28 April 2025

Digital Object Identifier 10.1109/MC.2025.3553360


Intelligence
Parallel Processing
Blockchain
COMPUTING edge
Software Evolution
Quantum Computing
Edge Computing
Blockchain
Hardware Security
High-Performance Chips
Automation
Machine Learning
Security and Privacy
Automation
Blockchain
Digital Transformation
Data Storage
Data Visualization
Social Media
Education
IEEE COMPUTER SOCIETY
IEEE