COMPUTING edge

- Security and Privacy
- Software
- Ethics
- Machine Learning

MARCH 2024

www.computer.org

IEEE COMPUTER SOCIETY

IEEE

COMPUTING edge

IEEE COMPUTER SOCIETY computer.org

IEEE COMPUTER SOCIETY

IEEE

Printed with inks containing soy and/or vegetable oils

STAFF

Editor
Lucy Holden

Production & Design Artist
Carmen Flores-Garvey

Periodicals Portfolio Senior Managers
Carrie Clark and Kimberly Sperka

Periodicals Operations Project Specialist
Christine Shaughnessy

Director, Periodicals and Special Projects
Robin Baldwin

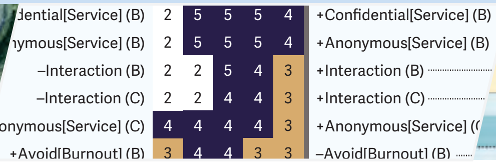
Senior Advertising Coordinator
Debbie Sims

Circulation: *ComputingEdge* (ISSN 2469-7087) 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.

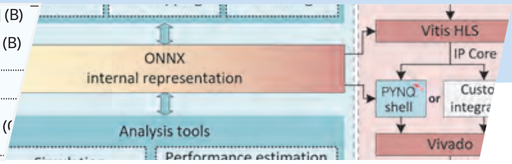
Postmaster: Send address changes to *ComputingEdge*-IEEE Membership Processing Dept., 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Printed in USA.

MARCH 2024 · VOLUME 10 · NUMBER 3

COMPUTING edge



dential[Service] (B)	2	5	5	5	4	+Confidential[Service] (B)
ymous[Service] (B)	2	5	5	5	4	+Anonymous[Service] (B)
–Interaction (B)	2	2	5	4	3	+Interaction (B)
–Interaction (C)	2	2	4	4	3	+Interaction (C)
nonymous[Service] (C)	4	4	4	4	3	+Anonymous[Service] (
+Avoid[Burnout] (B)	3	4	4	3	3	–Avoid[Burnout] (B)



Security and Privacy

Software

Ethics

Machine Learning

Departments

Magazine Roundup

The IEEE Computer Society's lineup of 12 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.

Computer

A Research Agenda for NFTs

Non-fungible tokens (NFTs) could potentially have a broader transformational effect than mere blockchain because they challenge the traditional notions of ownership and is, therefore, a more fundamental challenge to established economic and social structures. This article, in *Computer*'s December 2023 issue, provides a systematic review of the NFT literature outlining the research opportunities for NFTs.

computing in SCIENCE & ENGINEERING

Earth Virtualization Engines: A Technical Perspective

Earth Virtualization Engines (EVEs) aim to provide interactive and accessible climate simulations and data for a wide range of users. They combine high-resolution physics-based models with machine learning techniques to improve the fidelity, efficiency, and interpretability of climate projections. In this article, in *Computing in Science and Engineering*'s May/June 2023 issue, the authors summarize the technical challenges and opportunities for developing EVEs, and argue that they are essential for addressing the consequences of climate change.

IEEE Annals of the History of Computing

Computing Technologies for Resilience, Sustainability, and Resistance

The authors of this *IEEE Annals of the History of Computing* article from the October–December 2023 issue highlight some of the information and computing technologies (ICT) initiatives Māori have undertaken and showcase some of the real successes borne out over the past three decades, from Te Wahapū, translations of the Microsoft and Google, to the Niupepa, Māori Newspaper collection, to some of the work that is currently being done in the areas of xR, which encompasses augmented reality, virtual reality, and mixed reality, machine learning techniques, text-to-speech and speech-to-text conversion, and social media platforms.

IEEE Computer Graphics and Applications

Comparing Shape Representations for the Aesthetic Judgment of 3-D Shape Pairs

Visual aesthetics of 3-D shapes is a fundamental perceptual attribute. In this November/December 2023 *IEEE Computer Graphics and Applications* article, the authors explore the question of how different shape representations affect the aesthetic judgments of shape pairs. Their results have implications toward the data collection process of pairwise aesthetics data and the further use of such data in shape aesthetics and 3-D modeling problems.

IEEE Intelligent Systems

Whom to Trust, How and Why: Untangling Artificial Intelligence Ethics Principles, Trustworthiness, and Trust

The authors of this November/

2469-7087/24 © 2024 IEEE

December 2023 *IEEE Intelligent Systems* article present an overview of the literature on trust in artificial intelligence (AI) and AI trustworthiness and argue for distinguishing these concepts more clearly and gathering more empirical evidence on what contributes to people's trusting behaviors. AI systems should be recognized as sociotechnical systems, where the people involved in designing, developing, deploying, and using the system are as important as the system for determining whether it is trustworthy.

IEEE Internet Computing

A Tale of Two Cities: Data and Configuration Variances in Robust Deep Learning

In this article in the November/December 2023 issue of *IEEE Internet Computing*, the authors take a holistic view of deep neural network (DNN) robustness by summarizing the issues related to both data and software configuration variances. They also present a predictive framework using search-based optimization to generate representative variances for robust learning, considering data and configurations.

IEEE micro

On-Device Customization of Tiny Deep Learning Models for Keyword Spotting With Few Examples

Designing a customized keyword spotting (KWS) deep neural network (DNN) for tiny sensors is a time-consuming process, demanding training a new model on a remote server with a dataset of collected keywords. The authors of this November/December 2023 *IEEE Micro* article investigate the effectiveness of a DNN-based KWS classifier that can be initialized on-device simply by recording a few examples of the target commands. At runtime, the classifier computes the distance between the DNN output and the prototypes of the recorded keywords.

IEEE MultiMedia

Encoding of Media Value Chain Processes Through Blockchains and MPEG-21 Smart Contracts for Media

The authors of this October–December 2023 *IEEE MultiMedia* article describe the combination of the current set of MPEG-21 multimedia framework standards with distributed ledger technologies and smart contracts. Their gathering shapes the smart contracts for media, a specification that can be used to encode the terms and conditions of a contract for media-related delivery and consumption.

IEEE pervasive COMPUTING
MOBILE SYSTEMS | UBIQUITOUS COMPUTING | INTERNET OF THINGS

CityOutlook+: Early Crowd Dynamics Forecast Through Unbiased Regression With Importance-Based Synthetic Oversampling

This article, in *IEEE Pervasive Computing*'s October–December 2023 issue, studies crowd dynamics forecast one week in advance to detect irregular urban events, which plays an important role in infection prevention and crowd control. The authors propose an unbiased regression using importance weighting (IW), called CityOutlook, which successfully reduced the model bias and showed promising results. However, the straightforward weighting of the scarce data risks leading to the instability of the model due to the increase in model variance. To address this issue, the authors propose a nontrivial extension of their prior work called CityOutlook+ that realizes unbiased and less-variant regression by performing synthetic minority oversampling based on the importance.

IEEE SECURITY & PRIVACY

Journey to the Center of Software Supply Chain Attacks

The authors of this *IEEE Security & Privacy* article, in the November/December 2023 issue, discuss open source software supply chain attacks and propose a general taxonomy describing how attackers conduct them. They then provide a list of safeguards to mitigate such attacks. They also present their tool Risk Explorer for Software Supply Chains to explore such information, and then discuss its industrial use-cases.

IEEE Software

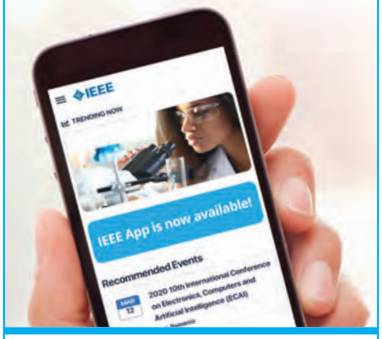
Explaining Black Boxes With a SMILE: Statistical Model-Agnostic Interpretability With Local Explanations

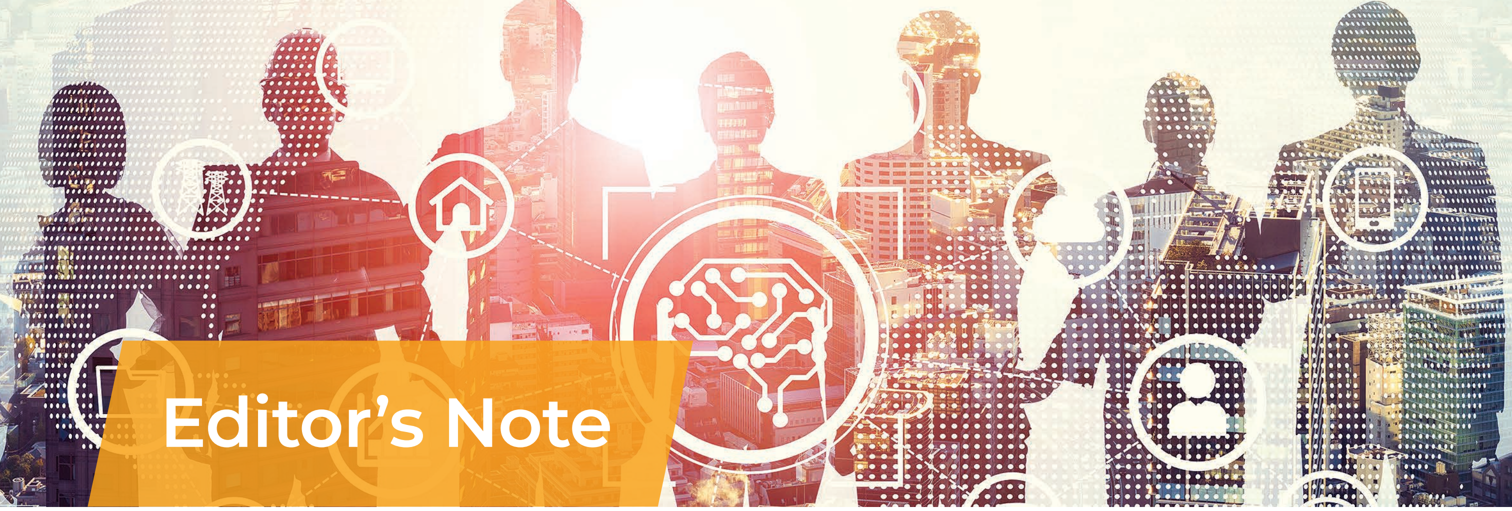
Explainability is a key aspect of improving trustworthiness. The authors of this January/February 2024 *IEEE Software* article therefore propose SMILE, a new method that builds on previous approaches by making use of statistical distance measures to improve explainability while remaining applicable to a wide range of input data domains.

IT Professional

Ransomware Attacks of the COVID-19 Pandemic: Novel Strains, Victims, and Threat Actors

During the COVID-19 pandemic (2020–2022), a significant disparity emerged between the demand for IT support services to sustain remote IT operations and the level of security controls contingent for maintaining uninterrupted and secure services over various digital platforms. Ransomware attacks increased by 150–200% through this time, causing disruptions in a wide range of industries. In *IT Professional*'s September/October 2023 issue, the authors of this article present a holistic analysis of popular ransomware attacks occurring during the pandemic and identify popular and novel ransomware strains that impacted businesses.


IEEE
TRENDING NOW
IEEE App is now available!
Recommended Events
2020 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)



Security and Privacy Risks and Solutions

Over the past few decades, individuals, businesses, and governments have transitioned from storing their records and sensitive information on paper, to storing them online. While physical records still pose security risks, the risk to online records is on the rise. This issue of *ComputingEdge* highlights the increasing impacts of cyberattacks and the need for better security in software development and automation. Additionally, this issue's articles emphasize the underutilized value of applying ethics to engineering as well as potential applications for machine learning.

Cyberattacks are becoming more prevalent and extreme worldwide. The authors of "The Impact of Cyberattacks on Small States," from *IEEE Software*, investigate the effects of cyberattacks on small states and propose possible AI defense strategies. *IT Professional*'s "Ransomware as a Business (RaaB)" delves into the high financial, productivity, and reputational costs that follow ransomware attacks.

Intensifying threats posed by stealthier online criminals lead businesses to question whether to trust their software supply chain. The authors of "Trusting Trust: Humans in the Software Supply Chain Loop," from *IEEE Security & Privacy*, explore the risks of relying on software built with open source infrastructure. *Computer*'s article "Placing Trust in Automated Software Development Processes" weighs the pros and cons of relying on automation to create dependable and secure code.

Incorporating ethics into technological development could help prevent software problems and pave the way for better lives. In *IEEE Software*'s "The Engineering Mindset Is an Ethical Mindset (We Just Don't Teach It That Way… Yet)," the authors explain how teaching ethics to computer scientists and engineers can help them craft more ethical programs. "Toward an Ethical Framework for Smart Cities and the Internet of Things," from *IEEE Internet Computing*, describes how smart cities built on ethical foundations can improve residents' technological, business, and interpersonal interactions, thus improving their overall quality of life.

The issue concludes with a discussion around machine learning. *IEEE Micro*'s article "RadioML Meets FINN: Enabling Future RF Applications With FPGA Streaming Architectures" outlines the possibility that deep neural networks (DNNs) will replace conventional radio signal processing as well as the associated risks. "Hybrid Models That Combine Machine Learning and Simulations," from *Computing in Science & Engineering*, indicates the benefits of using machine learning to enhance modeling and simulation techniques.

2469-7087/24 © 2024 IEEE

EDITORS: Michiel van Genuchten Straumann, genuchten@ieee.org
Les Hatton, Oakwood Computing Associates, lesh@oakcomp.co.uk

This article originally appeared in IEEE Software vol. 40, no. 4, 2023

DEPARTMENT: IMPACT

The Impact of Cyberattacks on Small States

Kristel M. de Nobrega, *Centrale Bank of Aruba*

Anne-Françoise Rutkowski and Piet Ribbers, *Tilburg University*

FROM THE EDITOR

The "Impact" series has often emphasized the importance of size and volume to survive in software and IT. But what if the size of your country is small and you face the same cyberthreats as much larger countries in the world? That is the case with small states that face the same cyberattacks while often having less means to defend themselves. You cannot grow a small state into a large state just to be able to defend yourself better against cyberattacks. What you can do is explained in this column. —*Les Hatton and Michiel van Genuchten*

At the time of writing this column, Russia and Ukraine are at war. Cyberattacks are part of the weapon arsenal in use. Cyberattacks have been launched on the central Bank of Poland targeting distributed denial of service (DdoS). In parallel, Israel reports its largest DdoS to date hitting government websites, making them unavailable. Also, the hacktivist group Anonymous is threatening to release proof related to a breach of the Russian Central Bank. The danger of escalating global conflicts in cyberspace is a hard reality. The National Atlantic Treaty Organization (NATO) has pronounced that a serious cyberattack on any of its members would trigger collective defense under Article 5. In a time of escalating tension with Russia and China, small states are in a difficult situation. Small states, such as the Pacific small island states, European landlocked countries, Baltic states, and the Caribbean region small open economies represent about a quarter of World Bank members. Those in the Asia–Pacific basin lack cyber forensic capability to gather enough evidence to substantiate geopolitically sensitive attribution. Mostly, even when attribution could be made, small states will choose peace over war, as market trade is essential to their survival and offense is not on their agenda. Small states in the Caribbean region are a perfect "sandbox" that enables attackers to test in an isolated setting the orchestration of their malicious activities.

In November 2019, a ransomware attack was launched on the only hospital in Aruba. The digital patient information systems became inaccessible, forcing the staff to fall back on to a manual system to ensure patients' care. In St. Martin, a black byte ransomware attack was launched on the national electrical grid locking out computers. The attackers paralyzed billing to customers and generated disconnections due to defaulters. In the financial sector, the Pan American Life Insurance Group operating from the United States and in the Caribbean got hit by a REvil ransomware attack.[1] Claims found on the dark web amount to 170 GB of stolen files as a result of the breach. In 2021, Microsoft's

Digital Object Identifier 10.1109/MS.2023.3265130
Date of current version: 14 July 2023

Digital Crimes Unit seized the websites of think tanks and human rights organizations of 29 countries (including Barbados, Dominican Republic, Jamaica, Trinidad, and Tobago).[2] Microsoft concluded that these websites would serve as launching base for intelligence-gathering purposes by the China-based hacking group *Nickel*.

TABLE 1. Observed cyber offense in the last 6 months from least (1) to most (5) frequently observed (CISG).

Attack class	Median	Mode	% Most frequent score
Probing attack	5	5	53.8%
Denial of service	2	2	7.7%
Remote to local (user) attacks	2	2	7.7%
User to root attacks	1	1	7.7%
Payload attacks	3	1–4	23.1%

The current costs of initiating cyberattacks seem to be lower than the cost of incident response and remediation. For example, costs to conduct an advance persistent threat (APT) sophisticated attack have been estimated between US$65,000 and US$542,000.[3] The cost to clean up is reported for 2022 to be US$4.35 million. One of the most dangerous traits of APT is the ability to run the background process silently, for example placing secured unnoticed back doors. The path to attack takes longer, as the aim of the adversary is to have long persistence. The high level of disguise and sophistication of APT make it difficult for organizations to notice. Hence, costs will only increase through time. APTs have been attributed to nation states with aggressive cyber defensive and offensive capabilities.

Small states experience more limitations in building capacity and developing cyber capability compared to nation states. Cybersecurity capacity is linked to variables, such as gross domestic product (GDP), that form a proxy for the available resources or cybersecurity capacity. Indeed, economies of scale and the availability of more financial and personnel resources allow deploying more controls, and therefore enhance the organizational capability in protecting the organization, hence the level of security maturity. Aruba has requested help from the Estonian e-Governance Academy (EGA 2022). Estonia became a leader in cyberdefense when it bounced back from its infamous cyberattack in 2007. It became a driving force in the European Union, proposing an integral national cyber strategy in 2008.[4] Estonia has a small GDP of approximately €187 billion and spends the NATO recommendation of at least 2% of their GDP on defense. This is nearly the entire GDP of a small state, such as Aruba.

INSIGHT INTO SMALL STATES PERCEPTION OF CYBER DEFENSE AND OFFENSE

Thirteen information security leaders from the Cybersecurity Information Sharing Group (CISG) bounded to the Caribbean region reported in a survey the frequency of cyber classes of attacks observed in the last 6 months (Table 1).

Also, we interviewed six chief information security officers (CISO) operating on the main critical infrastructures of Aruba. They mostly converge with the idea that a cyber defender in Aruba should know more about the "whole cyber picture" than a cyber expert in a larger state "who would have the luxury to know 'only' a small part in a particular area." The cyber competition is an offense-dominant clash. Particularly, when attackers and defenders are given equal resources, the attacker will usually prevail.[5] Attackers favor the offense because it offers anonymity, preventing meaningful deterrence.[6] For small state and for small enterprise, an offensive posture seems particularly challenging, not to say a "nonoption." One CISO commented on the stark lack of resources on the island in term of resources or cyber talent during the interview. Lack of technological and human resources is a major argument to rely further on security software and hardware on the island (e.g., sandbox, network monitoring, honeypot). Such technologies facilitate a data-driven approach in detecting more cyber threats, reducing de facto human intervention and bias.

CAN ARTIFICIAL INTELLIGENCE BE OF HELP TO CYBER EXPERTS?

It makes sense to investigate how artificial intelligence (AI) can contribute processing the data generated in cyberdefense. In 2023, it seems that AI is

TABLE 2. The future of AI according to OSSAT members, rated on a scale from strongly disagree (1) to strongly agree (7).

#	Quotations	Mean	Standard deviation	Mode
1	Model learning and resourcing takes time with current AI technology.	5.29	1.20	6
2	AI systems' learning may be the next target.	5.24	1.40	6
4	The human ability to improvise will remain important.	4.58	0.64	5
5	A broad hybrid combination of humans alongside AI agents and ethics are important to combat the hackers.	4.45	0.72	5
6	AI agents will be supportive to my work, not take over it.	4.34	0.71	4
7	AI agents will make us adapt to new ways of working.	4.32	0.62	4
8	Human input will remain key the coming years.	4.16	0.68	4
9	More people will be working with algorithms in the next coming years.	4.08	0.75	4
10	The human ability to make quick shots will remain key.	4.08	1.1	5
11	AI agents will evolve into a strategic technology for security specialists.	4.08	0.85	4
12	AI agents will not replace ethics.	4.05	1.06	4
13	Within 10 years there will be more autonomous AI agents taking over human tasks.	3.97	0.79	4
14	In the future it will be AI agents attacking against AI agents defending the organization.	3.66	0.94	4
15	AI agents will require a new edge of reasoning we may not be prepared for yet.	3.29	0.77	3
16	In the coming years some countries may be putting AI agents in jail.	1.92	0.88	2

supposed to aid in finding the solution to all problems in the world. This research started years back and reality is more stubborn. One CISO indicated that technical controls "are not a one-stop solution for cybersecurity." The interviewee emphasized that the speed of which vulnerabilities are being exploited by attackers is accelerating. Hence, specific proactive actions are required in, for example, deploying extra technologies to defend the organization. As another participant stated, "cybercriminals are becoming so advanced that you have to check more than in the past, when it was just checking the virus in the virus database. Right now, they are using all other strategies that the behavior needs to check, and certain triggers need to go through … more of what is going on." Another participant confirmed that, a "proactive approach should happen before the project goes live. Hence the human will be the last one holding the key and taking decision regarding security."

Thirty-eight members of the Operational Security Situational Awareness Teleconference (OSSAT) rated statements regarding the future of AI. [OSSAT has been organized by the European Central Bank since 2012 for sharing information on cybercrime in the financial services sector, vulnerabilities, technological trends and threats, and security incidents. The membership is limited to members of the Bank for International Settlements, international financial institutions (International Monetary Fund, World Bank), members of the European System of Central Banks, and the Computer Emergency Response Team for the European Union institutions, bodies, and agencies.] These 16 statements were originally collected via a focus group interview of cyber experts in the financial sector on the island Aruba.[7] Results are presented in Table 2, from highest to lowest score of agreement.

The top quote in Table 2 (#1) relates directly to the struggle small states face when lacking resources and time to defend. Information security specialists perceive the role of AI agents versus human rather positively (quotes #4, #6, #7, #8, #9, #10, #13). Still, there is little hope that AI and data fusion will leave a great space in terms of improvisation in cyberspace to human. Participants agree that a broad hybrid combination of humans alongside AI agents and ethics is important to combat hackers, and that humans will not be substituted by AI ethics (quote #5). For example, AI systems could be a support for forensic analysis that is required but lacking in small states, such as Aruba. AI can help in predicting the occurrences or reoccurrences of actual or potential criminal offences based on profiling of natural persons, based on a collection of past criminal behavior. The idea that in the future AI agents would end up in jail (quote #16) belongs probably to science fiction. A European Union proposal aims at extending a specific legal status to machines, holding these systems legally responsible for their actions.[8]

Outsourcing cyberdefense to Big Tech will, for more nations, entail a new form of legal sociopolitical challenge. AI holds a lot of promise for small states. In March 2023 (when we finished writing this column), *Wired* announced that "Microsoft's 'Security Copilot' Unleashes ChatGPT on Breaches."[9]

What may the future bring? The Data Breach Investigation Report in 2022 shows that 82% of breaches involved a human element.[10] Education, collaboration, and organization are key in the fight against cyberattacks, also for small-states. Interestingly, the majority of the stakeholders mentioned unity of effort to be important to defend properly. One CISO emphasized the importance of having this principle sorted out prior to an island-wide cyberattack, as "there should be an entity or body that would take the decision at that moment to decide who goes first in order of assistance if all are hit together." Security experts on the island believe that AI will aid cyberdefense professionals. Madnick stated, "The good guys are getting better, but the bad guys are getting badder faster."[11] As we indicated before in the series of "Impact" columns: "The benefits that legitimate developers enjoy are exactly the same for people who want to use software for criminal purposes."[12] The rat race is still on, with another tool in the arsenal of the good and the bad guys. One more example: When asking ChatGPT to generate some malware, it will first provide a politically correct answer. However, in February 2023, it was already reported that cybercriminals bypass ChatGPT restrictions using the openAI API.[13] The business model is already available, with some free queries, after which the price is an amount of money per 100 queries. The bad guys already figured out the integration of ChatGPT in their business model, while many legitimate companies have just started thinking about how to use ChatGPT in the first place.

The inherent limitations of small states, such as the Caribbean islands, with their focus on neutrality, metaphorically resembles fighting with wooden sticks against giants' elaborate attacks. What can we learn from the military, who have been in the defense business for a much longer time and consider cybersecurity very serious these days? The notion of *fighting power* has been applied to cyber defense.[14] Fighting power consists of three components: the *physical component* that relates to the "means to operate and fight," the *moral component* that relates to "people's will and ability to get people to operate and fight," and the *conceptual component* that addresses the "ideas behind how to operate."[14] The physical component comprises hardware and software, both virtual and physical assets, as well as information. So AI will help, but is only part of one of the three fighting power components. As Newton demonstrated, "a body in motion tends to stay in motion unless acted on by an outside force." Combining moral, conceptual, and physical components is crucial to reach a complex synergy to defend a system's moment of inertia. Small states should gain stability rather than being pushed around by external forces, such as attackers, expensive technological innovation, and abundance of legislation hard to cope with. Greater collaboration would serve as a major force, ensuring a greater stability of a small-island defensive system, hence, putting a strong break on the cyber rat race.

REFERENCES

1. "T&T among several countries hit by cyberattacks from international hacking group – Microsoft," *Guardian*, Dec. 2012. [Online]. Available: https://www.guardian.co.tt/news/tt-among-several-countries-hit-by-cyberattacks

-from-international-hacking-group--microsoft-6.2.1428159.21dafb6579

2. "Microsoft seizes sites used by APT15 Chinese state hackers," *Bleeping Comput.*, Dec. 2021. [Online]. Available: https://www.bleepingcomputer.com/news/microsoft/microsoft-seizes-sites-used-by-apt15-chinese-state-hackers/
3. *Hack at All Cost: Putting a Price on APT Attacks.* (2019). Positive Technologies Security. [Online]. Available: https://www.ptsecurity.com/ww-en/analytics/advanced-persistent-threat-apt-attack-cost-report/
4. N. Shafqat and A. Masood, "Comparative analysis of various national cyber security strategies," *Int. J. Comput. Sci. Inf. Secur.*, vol. 14, no. 1, p. 129, Feb. 2016.
5. A. F. Krepinevich, "Cyber warfare a 'Nuclear Option?' Cyber warfare," Center for Strategic and Budgetary Assessments, Washington, DC, USA, 2012. [Online]. Available: https://www.files.ethz.ch/isn/154628/CSBA_Cyber_Warfare_For_Web_1.pdf
6. R. Slayton, "What is the cyber offense-defense balance? Conceptions, causes, and assessment," *Int. Secur.*, vol. 41, no. 3, pp. 72–109, Jan. 2017, doi: 10.1162/ISEC_a_00267.
7. K. M. de Nobrega and A. F. Rutkowski, "The AI family: The information security managers best frenemy?" in *Proc. 55th Hawaii Int. Conf. Syst. Sci.*, 2022, pp. 184–193, doi: 10.24251/HICSS.2022.022.
8. "Laying down harmonized rules on artificial intelligence," European Commission, Brussels, Belgium, Apr. 2021. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021PC0206&from=EN
9. "Microsoft's 'Security Copilot' unleashes chatGPT on breaches," *Wired*, Mar. 2023. [Online]. Available: https://www.wired.com/story/microsoft-security-copilot-chatgpt-ai-breaches/
10. "Data breach investigations report." Verizon. Accessed: Feb. 19, 2022. [Online]. Available: https://enterprise.verizon.com/content/verizonenterprise/us/en/index/resources/reports/2022-data-breach-investigations-report.pdfdata-breach-investigations-report.pdf
11. S. Madnick, "Preparing for the cyberattack that will knock out U.S. power grids," *Harvard Bus. Rev.*, pp. 1–6, May 2017. [Online]. Available: https://cams.mit.edu/wp-content/uploads/2017-07.pdf
12. A. F. Rutkowski, M. van Genuchten, and L. Hatton, "No free lunch for software after all," *IEEE Softw.*, vol. 34, no. 5, pp. 13–15, Sep. 2017, doi: 10.1109/MS.2017.3571570.
13. "How-hackers-can-abuse-ChatGPT-to-create-malware," *Techtarget*, Feb. 2023. [Online]. Available: https://www.ghacks.net/2023/02/04/chatgpt-is-used-by-cybercriminals-to-write-better-phishing-emails/
14. P. A. Ducheine, J. van Haaster, and R. van Harskamp, "Manoeuvring and generating effects in the information environment," in *Netherlands Annual Review of Military Studies 2017: Winning Without Killing: The Strategic and Operational Utility of Non-Kinetic Capabilities in Crises*, P. A. L. Ducheine and F. P. B. Osinga, Eds. The Hague, The Netherlands: T.M.C. Asser Press, 2017, pp. 155–179.

KRISTEL M. DE NOBREGA is a manager of information security at the Centrale Bank of Aruba, Oranjestad, Aruba. Contact her at k.denobrega@cbaruba.org.

ANNE-FRANÇOISE RUTKOWSKI is a full professor in management of information at the Tilburg School of Economics and Management, Tilburg University, 5037 Tilburg, The Netherlands. Contact her at a.rutkowski@tilburguniversity.edu.

PIET RIBBERS is an Emeritus professor of information management, dean of the School of Economics, and head of Department of Information Systems and Management at Tilburg University, 5037 Tilburg, The Netherlands Contact him at p.m.a.ribbers@tilburguniversity.edu.

EDITOR: Jeffrey Voas, j.voas@ieee.org

DEPARTMENT: CYBERSECURITY

This article originally appeared in IT Professional vol. 24, no. 2, 2022

Ransomware as a Business (RaaB)

Nir Kshetri, University of North Carolina at Greensboro, Greensboro, NC, 27412, USA

Jeffrey Voas, IEEE Fellow

Ransomware attacks are growing rapidly. In this paper, the authors look at different types of costs to victims of such attacks.

According to Verizon, in 2020, ransomware accounted for 30% of all U.S.-based cyberattacks, which was more than double the rate for the rest of the world.[1] In a survey conducted among managed service providers, which remotely manages their clients' information technology (IT), 59% of the respondents reported that COVID-19-led remote working, increased exposure to cyberattacks and resulted in increased ransomware attacks.[2] A survey conducted among 300 U.S.-based IT decision-makers by cybersecurity company ThycoticCentrify, which has been published in its "2021 State of Ransomware Survey & Report," released in October 2021, found that 64% had been victims of a ransomware attack in the last 12 months, and 83% of those attack victims paid ransom.[3]

Ransomware attacks result in substantial economic costs. High-profile cases of ransomware payments to cybercriminals have attracted media attention. However, payments made to extortionists are only a small part of the story. The British cybersecurity company Sophos suggested that average ransomware recovery costs was $1.85 M USD in mid-2021 compared to $761,106 USD a year before.[4] Ransomware costs are estimated at $20 B USD in 2021 and expected to reach $265 billion in 2031 (see Figure 1).

COSTS TO VICTIMS

According to Sophos, the average ransom paid was $170,404 USD in 2020.[6] A conservative estimate is that ransomware criminals received $412 M USD in payments in 2020 (Table 1).[7]

Until recently, double extortion was ransomware criminals' strategy (asking organizations to pay for the decryption key to unlock the affected files and servers plus additional payments to destroy stolen data).[8] Several ransomware organizations also have dedicated leak sites to publish data stolen from victim organizations if they refuse to pay.[9] A newer extortion scheme was added in 2020: *triple extortion*. Here, criminals demand payments from the attacked organization's customers and third parties.

Remediation after successful attacks involves substantial costs. According to Proofpoint, the remediation process takes an average of 32,258 h for an average-sized organization with a total cost of more than $2 M USD (considering $63.50 USD hourly wage).[10] These costs are further increased due to criminals who fail to honor their promise to give decryptors to the victims or the decryptors do not work. Some decryptors do not work well with large files and others cannot handle large numbers of files. Some decryptors malfunction.[11] According to Trend Micro, about 33% of victim organizations that paid ransoms failed to get their data back.[12]

Productivity is severely affected.[13] A survey found that the average downtime due to ransomware is 21 days.[14] Employee productivity losses associated with the downtime was $3.2 M USD in 2021 compared to $1.8 M USD in 2015.[10]

Surprisingly, victims may face risks of regulatory fines and class action lawsuits.[10] Cloud provider Blackbaud faced 23 putative consumer class action lawsuits that were related to the ransomware attack that it faced in May 2020.[15] Among them, 17 were filed in the U.S. federal courts, four in the U.S. state courts, and two in Canadian courts. The attack affected more than 120 organizations.[16] Moreover, ransomware groups have been sanctioned by the U.S. Treasury Department's Office of Foreign Assets Control. If a ransomware victim pays to extortionists that are blacklisted by the U.S. Treasury Department, they could face fines of up to $20 M USD.[17]

Human resource management related costs are also significant. In a survey conducted by Sophos, over one-third of ransomware victims reported recruiting and retaining skilled cybersecurity professionals was the single biggest challenge compared to 19% of those that had not been attacked.[18]

1520-9202 © 2022 IEEE
Digital Object Identifier 10.1109/MITP.2022.3157208
Date of current version 6 May 2022.

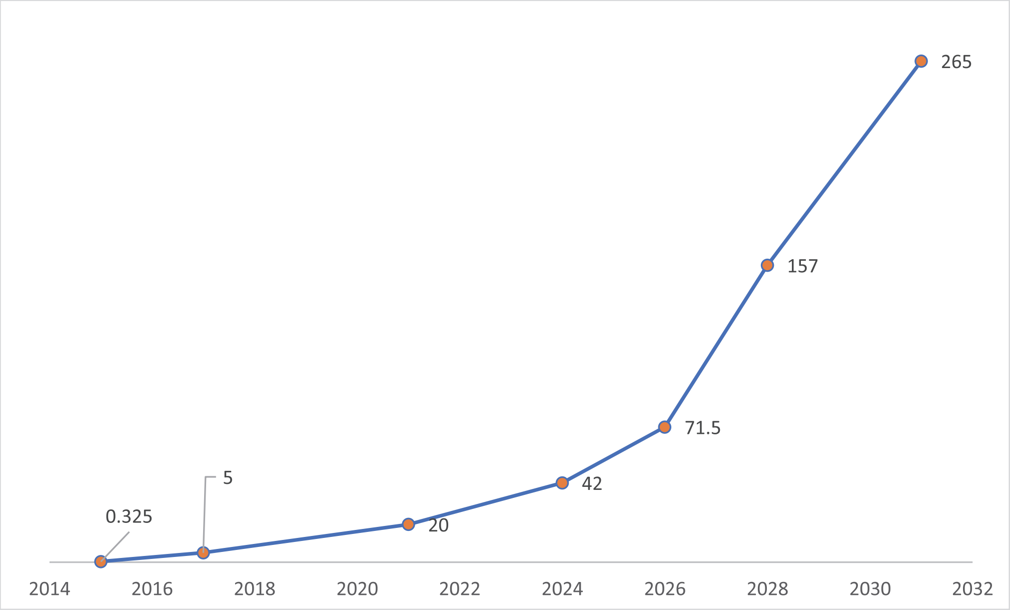


FIGURE 1. Economic costs of ransomware attacks ($, billion). Data source.[5]

Service disruption, failed transactions, and an inability to access information can lead to negative customer experiences. Organizations that cannot adequately protect personal data are less likely to be trusted. These factors contribute to the risk of customers switching to competitors.

Customers may lose trust after knowing that their data are at risk. According to Cybersecurity Ventures, over 66% of respondents would switch service providers if a provider failed to restore systems and applications within three days following. Over a third would switch after 24 h.[19] When customers switch, victim organizations experience a decrease in revenue.[10] A survey conducted by cybersecurity technology company Cybereason, which was published in the report titled Ransomware: The True Cost to Business, found that 53% of organizations victimized by ransomware suffered brand and reputation damage due to ransomware attacks. The survey also found that 66% of victim organizations experienced significant loss of revenue.[20]

Finally, there are also costs associated with the possible losses from the risks associated with future cyberattacks. Some victim organizations do not address the underlying issues that led to the initial attack. It is also possible that the perpetrators create backdoors that allow for continued access to the system after resolving the first attack. Attackers can also control malware to launch other attacks.[21]

Cybereason's survey of 1,263 cybersecurity professionals conducted in the U.S., United Kingdom, Spain, Germany, France, United Arab Emirates, and Singapore found that 80% of organizations that paid ransom faced repeat attacks.[22] Another study found that about 46% of the victims attacked were from the same perpetrator.[21] The U.K.'s National Cyber Security Centre detailed a case of a ransomware victim, who faced a repeat attack.[23] After facing an attack, the company paid cybercriminals millions of British pounds. The company fell victim to the same ransomware gang in less than two weeks later.[24]

MODUS OPERANDI

Most ransomware criminals use phishing. In 2021Q2, phishing accounted for 42% of ransomware attacks. Here, cybercriminals send emails containing a malicious attachment or direct victims to infected websites that attach ransomware.[25] When the receiver opens and attachment or visits the website, the files in the entire network are encrypted so that the victim company cannot access them.

Another popular modus operandi is an attack against Remote Desktop Protocol (RDP) services. Such attacks accounted for 42%.[24] Cybercriminals gain access to legitimate login credentials by phishing, guessing, or stealing. According to cybersecurity company ESET, there was a 768% growth of RDP attacks during 2020Q1–2020Q4.[26] ESET detected 29 billion attempted RDP attacks in 2020 as cybercriminals tried to exploit remote workers.

TABLE 1. Different costs associated with ransomware attacks.

Cost component	Explanation	Examples/statistics
Ransom payment	• Double/triple extortion schemes	• A conservative estimate: criminals received $412 million in 2020
Remediation	• Containing and cleaning up the malware and mitigating a vulnerability or a threat	• Proofpoint: total cost of more than $2 million
Downtime and lost productivity	• Negative effect on productivity	• Employee productivity losses: $3.2 million in 2021
Regulatory and legal costs	• Regulatory fines and class action lawsuits	• Blackbaud faced 23 consumer class action lawsuits
		• Fines for paying to extortionists blacklisted by the U.S. Treasury Department
Human resource management	• Challenges in recruiting and retaining skilled cybersecurity professionals	• Sophos survey: ransomware victims reported that they found it more difficult to recruit and retain skilled cybersecurity professionals
Brand and reputation damage	• Negative customer experience	• Cybersecurity ventures survey: risk of customers switching to competitors
		• Cybereason's survey: victims suffered brand and reputation damage and loss of revenue
Possible losses from the risks associated with future cyberattacks	• Victims' failure to address the underlying issues	• Cybereason: 80% of organizations that paid ransom faced repeat attacks.
	• Perpetrators create backdoors to use for continued access	
	• Control of malware in other parts of the network	

After breaching a network, cybercriminals may remain undetected in the system for a long time. Sophos found that the median time cybercriminals were in the victim company's network before releasing ransomware was 11 days. According to the Sophos Rapid Response team, the longest intruder dwell time was more than 15 months. This allows criminals to engage in malicious activities, such as lateral movement, reconnaissance, credential dumping, and data exfiltration.[27]

A key tactic that cybercriminals use to maximize harm and ensure compliance by victims is *lateral movement*.[28] This tactic involves moving deeper into a network to search for sensitive data and other high-value assets.[29]

Another tactic is *reconnaissance*, which is a preparation tool that entails engaging with the targeted system to gather information about the target before launching an attack. The goal is to identify weak points of the targeted system and devise an effective attack plan.[30] It is reported that for publicly traded companies, cybercriminals know annual revenues; this information is then used for how much ransom to demand. The ransomware-as-a-service group Conti looks at social media sites, such as LinkedIn, to identify the roles of employees and users that have privileged access. They are reported to be familiar with corporate network environments and where the most valuable assets are located and how such assets can be accessed.[31]

In a tactic known as credential dumping, criminals extract or "dump" user authentication credentials (usernames and passwords) from a compromised machine. Criminals often pull multiple passwords from a single machine, each of which can be used to access other computers on the network.[32]

Criminals also engage in data exfiltration to transfer sensitive data from the victim organization to their own systems.[33] One study found that data exfiltration occurred in 70% of all ransomware attacks in Q4 2020.[34]

FIGHTING BACK

Organizations can reduce the probability of being victimized by performing vulnerability assessments. Penetration testing can help. A robust firewall installed as a first line of defense can help monitor incoming and outgoing traffic and detect signs of malicious activities. However, firewalls must be regularly updated. It is reported that many companies are still using obsolete firewalls that were designed to tackle the cybersecurity threats of the mid-2000s.[35]

As noted, phishing is the source of many ransomware attacks. Phishing awareness training is important.

If employees click on a phishing link, they need further training.[36]

Attacks involving RDP can be prevented with mechanisms to protect passwords. They include increasing the quality of a password using a combination of the number and types of characters (e.g., lowercase, uppercase, numeric, and special characters), resetting them regularly, and not writing them down.

With network segmenting, it is possible to prevent hackers' lateral movement. Each subsystem in the network needs to have individual security controls and separate firewalls. Ransomware criminals require time to break into subsystems.[21]

Organizations also need to design privileged access management that specifies who has access to and what.[37] Microsoft has recommended that organizations build a "closed loop" system for privileged access. The goal is to ensure that only trustworthy "clean" devices, accounts, and intermediary systems, such as data warehouses or data repositories, have privileged access to sensitive systems and information. Ideally, the ability to perform privileged actions should be limited to a few authorized pathways.[38] Organizations need to monitor the pathways to privileged access and regularly audit the outcomes so that any leaks can be stopped.

Finally, repeat victimization is more common than what you might think. In the U.K. victim discussed earlier, the company was reported to fail to examine the cause of the attack and take corrective actions.[24] It is unwise for victims that have already faced attacks to not take actions to fix underlying root causes.

CONCLUSION

Ransomware attacks inflict economic loss. Ransom payments account for only a small proportion of the total costs. The defense measures mentioned can reduce the chance of being victimized. However, ransomware appears to be a long-term problem with few quick or easy solutions.

DISCLAIMER

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

REFERENCES

1. G. De Vynck, R. Lerman, E. Nakashima, and C. Alcantara, "The anatomy of a ransomware attack," Washington Post, Jul. 9, 2021. [Online]. Available: https://www.washingtonpost.com/technology/2021/07/09/how-ransomware-attack-works/
2. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/2te98jpb
3. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/338vs2wt
4. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/mvfe7s96
5. Cybercrime Magazine, *Global Ransomware Damage Costs Predicted to Exceed $265 Billion by 2031*, 2021. [Online]. Available: https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/
6. D. Winder, *Ransomware Reality Shock: 92% Who Pay Don't Get Their Data Back*, 2021. [Online]. Available: https://www.forbes.com/sites/daveywinder/2021/05/02/ransomware-reality-shock-92-who-pay-dont-get-their-data-back/?sh=3ac82669e0c7
7. E. Nakashima, "U.S. aims to thwart ransomware attacks by cracking down on crypto payments," Washington Post, 2021. [Online]. Available: https://www.washingtonpost.com/business/2021/09/17/biden-sanctions-ransomware-crypto/
8. D. Carmack, "What we know about DarkSide, the Russian hacker group that just wreaked havoc on the east coast," The Heritage Foundation, 2019. [Online]. Available: https://www.heritage.org/cybersecurity/commentary/what-we-know-about-darkside-the-russian-hacker-group-just-wreaked-havoc
9. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/bdcpuz7k
10. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/5eptmzy6
11. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/y6e5zfyc
12. J. R. Stark, "Ransomware payment: Legality, logistics, and proof of life," 2019. [Online]. Available: https://listingcenter.nasdaq.com/assets/Ransomware_White_Paper_1.pdf
13. "Downtime: The real cost of ransomware," 2019. [Online]. Available: https://www.delphix.com/blog/downtime-real-cost-ransomware
14. Accessed: Dec. 15, 2021. [Online]. Available: https://jaxenter.com/ransomware-security-patton175425.html
15. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/mr33enbz
16. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/yavy2pb5
17. "Ransomware victims who pay up could face fines of up to $20m," 2020. [Online]. Available: https://www.finextra.com/newsarticle/36673/ransomware-victims-who-pay-up-could-face-fines-of-up-to-20m
18. "Cybersecurity: The human challenge." [Online]. Available: https://www.sophos.com/en-us/content/cybersecurity-the-human-challenge.aspx

19. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/yckceebm
20. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/2p8fzwb7
21. Accessed: Dec. 15, 2021. [Online]. Available: https://www.onsitecomputersinc.com/blog/ransomware/
22. "80% of ransomware victims suffer repeat attacks, according to new report," 2020. [Online]. Available: https://www.cbsnews.com/news/ransomware-victims-suffer-repeat-attacks-new-report/
23. "The rise of ransomware," The National Cyber Security Centre, 2020. [Online]. Available: https://www.ncsc.gov.uk/blog-post/rise-of-ransomware
24. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/4f8tb748
25. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/4w5dwb5p
26. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/28mtdzcw
27. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/2p84s8ss
28. "Lateral movement explained, what is lateral movement?," 2021. [Online]. Available: https://www.crowdstrike.com/cybersecurity-101/lateral-movement/
29. Accessed: Dec. 15, 2021. [Online]. Available: https://attack.mitre.org/tactics/TA0008/
30. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/yrvft2p3
31. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/2p8uw2cv
32. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/54w26dhk
33. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/34656bxb
34. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/4wj3bdhc
35. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/2p882cfe
36. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/5dy5fzbf
37. Accessed: Dec. 15, 2021. [Online]. Available: https://www.coresecurity.com/privileged-access-management
38. Accessed: Dec. 15, 2021. [Online]. Available: https://tinyurl.com/5b5zyyrb

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

JEFFREY VOAS is the editor-in-chief of *Computer*. He is the fellow of IEEE. Contact him at j.voas@ieee.org.

ADVERTISER INFORMATION

Advertising Coordinator

Debbie Sims
Email: dsims@computer.org
Phone: +1 714-816-2138 | Fax: +1 714-821-4010

Advertising Sales Contacts

Mid-Atlantic US, Northeast, Europe, the Middle East and Africa:
Dawn Scoda
Email: dscoda@computer.org
Phone: +1 732-772-0160
Cell: +1 732-685-6068 | Fax: +1 732-772-0164

Southwest US, California:
Mike Hughes
Email: mikehughes@computer.org
Cell: +1 805-208-5882

Central US, Northwest US, Southeast US, Asia/Pacific:
Eric Kincaid
Email: e.kincaid@computer.org
Phone: +1 214-553-8513 | Fax: +1 888-886-8599
Cell: +1 214-673-3742

Midwest US:
Dave Jones
Email: djones@computer.org
Phone: +1 708-442-5633 | Fax: +1 888-886-8599
Cell: +1 708-624-9901

Jobs Board (West Coast and Asia), Classified Line Ads

Heather Buonadies
Email: hbuonadies@computer.org
Phone: +1 623-233-6575

Jobs Board (East Coast and Europe), SE Radio Podcast

Marie Thompson
Email: marie.thompson@computer.org
Phone: +1 714-813-5094

FROM THE EDITORS

Trusting Trust: Humans in the Software Supply Chain Loop

Laurie Williams, *Associate Editor in Chief*

This article originally appeared in
IEEE SECURITY & PRIVACY
vol. 20, no. 5, 2022

> *To what extent should one trust a statement that a program is free of Trojan horses? Perhaps it is more important to trust the people who wrote the software. . . . You can't trust code that you did not totally create yourself.*
>
> —Ken Thompson, Turing Award Lecture, 1984[1]

The modern world relies on digital innovation in almost every human endeavor and for our critical infrastructure. Digital innovation has accelerated substantially as software is increasingly built on top of layers of reusable abstractions, including libraries, frameworks, and cloud infrastructure, which often lie outside an organization's trust boundary. Where previous teams of engineers invested months, today, beginners can write intelligent smartphone apps with a few lines of code. Leveraging these reusable abstractions gives rise to software supply chains, where software products include "upstream" components as well as dependencies, created and modified by others, that, again, often include their own transitive dependencies. Most of these dependencies are open source projects.

However, with all of the power that software supply chains and open source infrastructure provide also come risks. Software developers did not anticipate how the software supply chain would become a deliberate attack vector. The software industry has moved from passive adversaries finding and exploiting vulnerabilities contributed by honest, well-intentioned developers to a new generation of software supply chain attacks where attackers aggressively implant vulnerabilities directly into infrastructure software (e.g., libraries or tools) and infect build and deployment pipelines.

Sonatype[2] reports a 650% year-over-year increase in detected supply chain attacks (on top of a 430% increase in 2020) targeted toward upstream open source repositories. The U.S. government is so concerned about software supply chain security deficiencies that a whole section of Executive Order 14028[3] (*Improving the Nation's Cybersecurity*), issued on 12 May 2021, is focused on new compliance requirements for government vendors to enhance supply chain security.

Digital Object Identifier 10.1109/MSEC.2022.3173123
Date of current version: 6 September 2022

WHERE PREVIOUS TEAMS OF ENGINEERS INVESTED MONTHS, TODAY, BEGINNERS CAN WRITE INTELLIGENT SMARTPHONE APPS WITH A FEW LINES OF CODE.

Historically, when people thought about the software supply chain attack surface, they thought about the many components that make up a product. More recently, the software supply chain attack surface increasingly encompasses the build infrastructure. In this article, I bring back the progressive thoughts of Ken Thompson and place humans in the software supply chain—as both developers with and without malicious intent and as part of the solution to software supply chain security.

COMPONENTS AND THE SOFTWARE SUPPLY CHAIN

Attackers exploit vulnerabilities in components. For example, in late 2021, an accidentally injected vulnerability in the popular logging library log4j, used by more than 35,000 Java packages, allowed an attacker to perform remote code execution by exploiting an insecure Java Naming and Directory Interface (JNDI) lookup feature, which is enabled by default in many versions of the library. In 2022, as an instance of protestware, a developer maliciously injected code into the node-ipc package, with more than 700,000 weekly downloads. The initial version of the malicious code attempted to geolocate where the code is running, and, if it discovers it is running within Russia or Belarus, then it attempts to replace the contents of every file on the system with a Unicode heart character.

To manage the component-based supply chain risks, development teams (those humans!) are challenged to update their components when vulnerabilities are found and choose safe components.[4,5] Software composition analysis (SCA) tools aid in identifying vulnerable components. SolarWinds was a wakeup call that reminded security experts that quickly updating to the latest version of a dependency might also introduce malicious code or vulnerable code that may be exploitable. Projects such as Open Source Security Foundation (OpenSSF) Metrics and deps.dev are emerging to provide metrics on open source components to aid teams in making informed choices on components.

BUILD INFRASTRUCTURE AND THE SOFTWARE SUPPLY CHAIN

In an emerging attack vector, attackers are infiltrating the build infrastructure. In 2020, the build process for the SolarWinds network management tool, Orion, which is used to manage routers and switches inside corporate networks, was maliciously subverted to distribute malware to create backdoors on victims' networks. This malware enabled spying on at least 100 companies and nine U.S. government agencies, including the Centers for Disease Control and Prevention, U.S. Department of Homeland Security, U.S. Department of Justice, Pentagon, and U.S. Department of State.

In 2021, attackers used a mistake in how Codecov built docker images to modify a script, which allowed them to send the environment variables from the continuous integration (CI) environment of Codecov customers to a remote server. The attackers accessed private Git repositories from the Git credentials in the CI environment and exploited the secrets and data within.

To manage the build infrastructure-based supply chain risks, development teams (those humans!) are challenged to secure their build infrastructure, considered to be a huge open-ended challenge.[4] The Supply Chain Levels for Software Artifacts [SLSA (pronounced "salsa")] framework provides a checklist of standards for reasoning about the build process. SLSA is based on Google's internal processes and defines four levels, beginning with simply having a scripted build and recording provenance information and ending with using an ephemeral, isolated, parameterless, and hermetic build environment. Bonus points are given if the build is reproducible; i.e., two builds produce bit-for-bit identical output.

TO MANAGE THE COMPONENT-BASED SUPPLY CHAIN RISKS, DEVELOPMENT TEAMS (THOSE HUMANS!) ARE CHALLENGED TO UPDATE THEIR COMPONENTS WHEN VULNERABILITIES ARE FOUND AND CHOOSE SAFE COMPONENTS.

Additionally, the industry is increasingly moving toward the use of reproducible builds to verify that the source code was unaltered when the original build was produced. There are a number of efforts on this front. For example, the Debian-initiated https://reproducible-builds.org effort has characterized and classified the many types of nondeterminism that can be introduced during the build process.

HUMANS AND THE SOFTWARE SUPPLY CHAIN: ATTACKERS

In the supply chain, we can consider attackers as developers who act with malicious intent. Attackers aggressively implant vulnerabilities directly into components, infrastructure, software (e.g., libraries and tools) and infect build and deployment pipelines. Back to Ken Thompson's quote about trusting trust, "Perhaps it is more important to trust the people who wrote the software. . . . You can't trust code that you did not totally create yourself".[1] In reality, innovation would grind to a halt in an organization that decides it can't trust any open source code due to the risk of malicious code injection. That would be like Tesla deciding it can't trust its screw manufacturer and manufacturing its own screws.

As an industry, we need to develop models for identifying malicious actors and malicious code injection. Because the attackers act in ways that well-meaning developers do, we are challenged to identify their actions. Models are beginning to emerge to identify weak leaks signals that arouse suspicion, such as the identification that a component maintainer's domain is expired and does not have two-factor authentication (2FA) authentication set up on the account. An attacker can relatively easily hijack that component or a component that has an install script.[6]

More signals that indicate malicious activity need to be developed and verified. We can't stop the attackers, but we can make it harder for them. For example, typosquatting was a very popular attack vector. As ecosystems automated the identification and takedown of rogue typosquatted packages, attackers have moved away from this attack vector. However, we play "cat and mouse"—with the plethora of weaknesses in most applications and infrastructures, moving to a different spot on the attack surface is not a big deal for the attacker but can be a big deal for the defender.

> *WHILE DEVELOPERS MAY FEEL A POPULAR PACKAGE MUST BE SECURE, ATTACKERS INTENTIONALLY LEVERAGE THEIR EFFORTS BY INJECTING MALICIOUS CODE IN PACKAGES WITH MANY DEPENDENTS AND A HIGH DOWNLOAD FREQUENCY.*

HUMANS AND THE SOFTWARE SUPPLY CHAIN: SOFTWARE DEVELOPERS

> *Some might argue that it's almost too easy to introduce a new dependency into your software systems. I'm definitely guilty of this in my previous life as an engineer. I remember pulling in random Python packages when building my own websites and not putting any thought into security. It should be fine if so many other people are using the same package, right?*
>
> —Kim Lewandowski, Google Product Manager[7]

In the supply chain, we can consider software developers as well-intentioned actors in the supply chain who are just trying to deliver functionality but sometimes make mistakes that enable security breaches. The quote from Lewandowski epitomizes a common but now naive belief held by developers. While developers may feel a popular package must be secure, attackers intentionally leverage their efforts by injecting malicious code in packages with many dependents and a high download frequency. A popular package may, in fact, be more risky.

Predominantly measured by his or her ability to deliver functionality, a developer can be overwhelmed and overloaded by the additional compliance restrictions and the notifications from supply chain security tools. For example, SCA tools, such as Dependabot, send email and pull requests for every dependency and transitive dependency in a package that has a discovered vulnerability. The vulnerability may be in a part of a component not used by the package, and an automatic acceptance of the pull request may break functionality and/or pose additional security risk—increasing, not lowering, the overall risk.

Additionally, package maintainers may be overloaded, which may lead to hasty and possibly dangerous decisions around accepting new maintainers and pull requests. (They are humans, after all.) For example, a study on the npm ecosystem revealed that the top 1% of maintainers own an average number of 180.3 packages, with an average of 4,010 direct dependents.[6] That's a lot!

THE HUMANS AS FIRST-CLASS PLAYERS IN THE SECURE SOFTWARE SUPPLY CHAIN SOLUTION

For humans to be the solution to supply chain security, developers need education, guidance, and risk-based tools. Part of this education is just the awareness that not all open source software can be trusted. Major players in the industry are already coming together via a number of projects. Both SLSA (mentioned earlier) and the Open Web Application Security Project (OWASP) Software Component Verification Standard provide frameworks for identifying activities, controls, and best practices that can help in identifying and reducing risk in a software supply chain. Additional projects include OpenSSF (mentioned earlier); sigstore; and in-toto,[8] a joint industry–academia project that helps shed light on code-to-binary provenance. Package managers and researchers are exploring logic-based and machine learning-based mechanisms for identifying malicious code and malicious contributors. Currently, this machine learning-based sorting to identify bad hygiene has a high signal-to-noise ratio and presents technical challenges, so more work is needed.

Is it possible to trust trust? Can we develop mechanisms for software developers to trust code that we did not totally create ourselves in an informed manner?

ACKNOWLEDGMENTS

My thinking on the role of people in the software supply chain was highly influenced by conversations with 30 practitioners during three Software Supply Chain Summits. These were coheld by William Enck and me,[4] with extensive collaboration with Yasemin Acar, Michel Cuckier, William Enck, Alex Kapravelos, and Christian Kästner—especially Yasemin, who is one of the few researchers at the intersection of human factors and secure software development.

REFERENCES

1. K. Thompson, “Reflections on trusting trust,” *Commun. ACM*, vol. 27, no. 8, pp. 761–763, Aug. 1984, doi: 10.1145/358198.358210.
2. “2021 State of the software supply chain,” Sonatype, Fulton, MD, USA, Jul. 2021. https://www.sonatype.com/resources/state-of-the-software-supply-chain-2021
3. “Executive order 14028: Improving the nation’s cybersecurity,” Federal Register, May 12, 2021. https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nations-cybersecurity
4. W. Enck and L. Williams, “Top five challenges in software supply chain security: Observations from 30 industry and government organizations,” *IEEE Security Privacy*, vol. 20, no. 2, pp. 96–100, 2022, doi: 10.1109/MSEC.2022.3142338.
5. D. Drusinsky and J. Michael, “Obtaining trust in executable derivatives using crowdsourced critiques with blind signatures,” *Computer*, vol. 53, no. 4, pp. 51–56, Apr. 2020, doi: 10.1109/MC.2020.2970819.
6. N. Zahan, L. A. Williams, T. Zimmermann, P. Godefroid, B. Murphy, and C. S. Maddila, “What are weak links in the NPM supply chain?” in *Proc. Int. Conf. Softw. Eng. Softw. Eng. Pract.*, 2022, to be published.
7. K. Lewandowski, “Security scorecards for open source projects,” Open Source Security Foundation Nov. 6, 2020. [Online]. Available: https://openssf.lfprojects.linuxfoundation.org/blog/2020/11/06/security-scorecards-for-open-source-projects/
8. S. Torres-Arias, H. Afzali, T. K. Kuppusamy, R. Curtmola, and J. Cappos, “in-toto: Providing farm-to-table guarantees for bits and bytes,” in *Proc. USENIX Security Symp.*, Aug. 2019, pp. 1393–1410.

EDITOR: James Bret Michael, IEEE Senior Member, bmichael@nps.edu

DEPARTMENT: CYBERTRUST

This article originally appeared in Computer vol. 55, no. 7, 2022

Placing Trust in Automated Software Development Processes

James Bret Michael, *Naval Postgraduate School*

Automation of certain aspects of software development and maintenance help us achieve our software-productivity goals, but we need to consider the trust we can place in that automation.

IBM's release of the first commercial off-the-shelf compiler for a high-level programming language was a watershed event.[1] To a large extent, it freed programmers from hand-coding programs in assembly language or microcode, thereby significantly reducing the total number of source lines of code for a human to construct a program; the expansion ratio for that version of Fortran to assembly language was 1:20 for the IBM 704 computer. As compiler technology matured and became available for other high-level languages and computers, programmers' use of high-level programming languages flourished. This, in turn, was a major contributor to improved productiveness in maintaining software: it was relatively easy for humans to understand and modify the structure and function of legacy software presented in a high-level language compared to doing so for the equivalent assembly or microcode representation.

Major advances over the next three decades, for example, in compiler-optimization techniques, programming language features, such as user-defined functions with passing of values by reference and the capability for precise data description, and structured programming, along with the introduction of tools such as cross compilers and compiler–compilers, further liberated programmers from performing software development and maintenance tasks that are too time consuming or error prone for humans to accomplish but well suited for mechanization. In addition to reducing development time, programmers had the opportunity to use the resultant time savings afforded by applying those tools and techniques to make updates or devote additional attention to other tasks, such as performing software requirements engineering, architecting and designing software, and testing and debugging code.

I LOOK BACK TO 1980, THE TIME AT WHICH I FIRST EXPERIENCED SOME OF THE CHALLENGES ASSOCIATED WITH DEVELOPING AND MAINTAINING SOFTWARE PROGRAMS.

From the late 1980s through the 1990s, major strides were made in improving the engineering of software, for example, with the widespread adoption of object-oriented design and programming using languages such as Ada, software reuse, Barry Boehm's spiral model of software development, software metrics, Watts Humphrey's Capability Maturity Model, Java virtual machine and Java development kits, rapid system prototyping, and software–hardware co-design.[2,3] Some other examples include the rapid maturation and use of computer-aided software engineering (CASE) tools, toolchains, integrated development environments, software frameworks (which include items such as code libraries and support software), aspect-oriented programming, and

Digital Object Identifier 10.1109/MC.2022.3171885
Date of current version: 4 July 2022

Unified Modeling Language-based modeling and code generation.

During that time period, I was conducting applied research in software engineering at the Software Productivity Consortium and the Computer and Software Engineering Division of the Institute for Defense Analyses. With the sponsors of that research, we were attempting to obtain efficiencies in the development and maintenance of systems as the software partition (that is, the portion of a system implemented in software) of those acquired systems increased in size and complexity. New conferences were launched to help spur advances in software productivity and software engineering in general, such as the founding of the IEEE-sponsored First International Conference on the Software Process in 1991.[4]

Over the last two decades, our profession made further inroads in boosting software productivity through the use of capabilities such as automated tool support for distributed software development, agile development methods, cloud services, containerization, and the integration of development and operations (known as *DevOps*). We also concentrated on making best practices and the accumulated body of knowledge for software engineering widely available to software professionals. Productivity gains also have been realized by reverse engineering through the use of automation support, such as the open source and extensible Ghidra framework (https://ghidra-sre.org/).

I look back to 1980, the time at which I first experienced some of the challenges associated with developing and maintaining software programs. I wrote the software in WATerloo Fortran IV along with scripts in job control language, which included lots of write statements with the text 'got to here' for debugging purposes. I submitted the software to an IBM System/360 mainframe computer using 80-column punched cards and waited for what seemed then to be an eternity for my batch jobs to run and the line-printer output to be made available. I can attest that our profession has made major strides in supporting software development and maintenance, especially in terms of software productivity. And the "rubberband bandito," as we affectionately called the rubberband that someone left on a deck of cards that was placed in the card-input reader, which then inevitably became jammed in the reader mechanism, was a real productivity killer! However, I had nothing to complain about. In the early 1960s, my father was programming computers that had plugboards, switches, and vacuum tubes—an even more tedious and time-consuming process and one that required a detailed knowledge of the computer's hardware design and functionality.

However, the preceding example results in an apples-to-oranges comparison because the complexity and size of typical software applications today are many orders of magnitude beyond those of the 1980s. In addition, today's software processes bear little resemblance to those of the past and are now typically tailored to the type and attributes of the software application being developed, in addition to the policies, capabilities, and resources of the developers and maintainers of the software. Even the automation support for developing and maintaining software is bundled differently. For instance, software engineers and programmers typically use software development kits (SDKs) to develop and maintain mobile, web, desktop, and embedded software applications. An SDK consists of just about everything you need to build and update an app, including a compiler, an editor, code samples, a framework, an integrated development environment, application programming interfaces, testing and analytics tools, support documentation, and debuggers. SDKs have been shown to improve software productivity, as evidenced, for example, by shortened software development cycles for apps.

The picture I painted of the evolution of software productivity is incomplete as just about every advancement made in computer science, software engineering, and related fields, such as computer engineering, has had some effect on software productivity. In addition, I have not mentioned the many challenges encountered along the way in achieving improvements in software productivity. A good example from the 1980s is that CASE tools were not originally designed to support distributed development of software, nor did they afford the capability to merge system artifacts developed using multiple CASE tools into a single picture of a system—what we referred to at the time as the information modeling gap.[5]

It has also been challenging over the past six decades to convince software engineers and programmers to adopt new technologies (for example, pattern-based design of software) or modify their familiar software-process workflows to better leverage these technologies. I personally observed such resistance by project managers to adopt automated test-case generation and execution tools, even though there was no way for their technical teams to be successful at manually constructing a sufficient number of test cases and updating that set of test cases for the mission- and safety-critical software they were developing. Their excuses for reluctance to change came down to the perception that the project risk and the expense associated with the learning curve in using new technologies or ways of doing things were too high. Another confounding factor has been the ongoing challenge in gaining access to, collecting, analyzing, and publicly disseminating information about software development productivity from real-world software development projects.

ANOTHER CONFOUNDING FACTOR HAS BEEN THE ONGOING CHALLENGE IN GAINING ACCESS TO, COLLECTING, ANALYZING, AND PUBLICLY DISSEMINATING INFORMATION ABOUT SOFTWARE DEVELOPMENT PRODUCTIVITY FROM REAL-WORLD SOFTWARE DEVELOPMENT PROJECTS.

SOFTWARE PROCESS AUTOMATION AND TRUST

Much of what I have described about improving software productivity involves automation of portions of a software process. The push for enhancing the automation of software processes continues. For example, there is now commercial support for low-code and no-code development of software, with the aim of making it possible for people with little to no knowledge of computing to be able to develop and maintain software applications with some or no assistance from computing professionals.

As another example, software–hardware co-design has gained new momentum as developers of systems work to optimize the performance of both the software and hardware that run artificial intelligence (AI) and machine learning (ML) algorithms. In addition, ML operations is being leveraged by AI practitioners to gain productivity through automation support for the end-to-end development and maintenance of ML software (see, for example, https://ml-ops.org/).

As I pondered my and other computing professionals' increasing reliance on software process automation, I had a feeling of déjà vu. Ken Thompson's Turing Award lecture, "Reflections on Trusting Trust," provides an example of a compiler that, through self-reproducing programs, introduces and reintroduces security-related software bugs.[6] The takeaway, according to Thompson, is that "You can't trust code that you did not totally create yourself." Yes, the productivity gains that can be obtained from using compilers and other forms of automation for software development and maintenance can be substantial, but how much trust can we place in the dependability of the software (and in the case of software–hardware co-design, the software and hardware) artifacts that the automation produces?

From a practical standpoint, we are reliant on high levels of automation of the software development process to make it technically feasible to build modern-day applications. I do not know about you, but I do not have the interest, nor is it practical in terms of time and other resources, to develop and maintain all of my own automated support for enacting the software development processes that I use. Development and maintenance of software involves assuming risk by leveraging the products and services from local, regional, or global supply chains.

Consider the widely used Apache Log4j Java logging framework (https://logging.apache.org/log4j/2.x/), an example of improving productivity through reuse. An unintended software bug in Log4j happened to present a zero-day vulnerability (https://nvd.nist.gov/vuln/detail/CVE-2021-44228), which eventually was exploited. This had an impact on the trust software professionals have in Log4j, but I wonder how much trust was restored after that vulnerability was discovered. Shortly after that patch was released, an exploitable vulnerability was found in the patch, which, in turn, needed to be fixed (https://nvd.nist.gov/vuln/detail/CVE-2021-44832), further affecting trust in the logging utility. The discovery of additional flaws is not surprising—those of us who have practiced software reliability engineering know that software updates typically introduce software flaws.

We also continue to find previously unknown security, reliability, resilience, and other dependability-related issues in compilers and other enablers of software development process automation, such as the lack of adequate defenses against attacks that exploit Unicode Bidi overrides. Note that this override capability for use with text encodings is an intended feature, not a software bug.[7] When supply-chain-enabled attacks involve inserting untrusted code into software repositories, there are ways to mitigate the risk, even for executable derivatives and tools (for example Log4j), such as through the use of crowdsourcing critiques with blind signatures.[8]

There are legal and ethical aspects impacting the trust we can place in software development process automation, such as the revelation that the company known as Anomaly Six allegedly uses the SDKs inside certain mobile apps to collect information about the user of a mobile phone and fuse those data with geolocation data to track the movement of the person using the mobile phone.[9] This is not the first time users have been duped into sharing their personal information. Avast collected and sold information to third parties about the web-browsing habits of the users of Avast's security software.[10]

There are many other aspects of trust to be explored—too many to cover in this article. I plan to write a follow-on article for this column on accessing the suitability of a programming language for use in developing trusted software. However, let's end on an upbeat note, which is that companies are making progress toward devising frameworks, tools, and other software development process automation to support explainable AI. Helping developers, maintainers, users, and other stakeholders (such as regulators and policy makers) understand the behavior of ML algorithms and documenting that behavior will be key for obtaining trust in the dependability of AI-enabled systems. We also need to be mindful of managing the risk involved in using those explainable AI capabilities and further enhance our software development process automation to accommodate the current paradigm shift from model-centric to data-centric AI.

> *THE DISCOVERY OF ADDITIONAL FLAWS IS NOT SURPRISING—THOSE OF US WHO HAVE PRACTICED SOFTWARE RELIABILITY ENGINEERING KNOW THAT SOFTWARE UPDATES TYPICALLY INTRODUCE SOFTWARE FLAWS.*

DISCLAIMER

The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. government. The U.S. government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright annotations thereon.

REFERENCES

1. J. W. Backus *et al.*, "The FORTRAN automatic coding system," in *Proc. AFIPS 1957 Western Joint Comput. Conf.*, 1957, pp. 188–198. Accessed: Apr. 22, 2022. [Online]. Available: http://www.bitsavers.org/pdf/ibm/704/FORTRAN_paper_1957.pdf
2. B. W. Boehm, "A spiral model of software development

and enhancement," *Computer*, vol. 21, no. 5, pp. 61–72, 1988, doi: 10.1109/2.59.

3. W. S. Humphrey, "Characterizing the software process: A maturity framework," *IEEE Softw.*, vol. 5, no. 2, pp. 73–79, 1988, doi: 10.1109/52.2014.
4. *Proceedings of the 1st International Conference on the Software Process*, Los Alamitos, CA, USA: IEEE Computer Society Press, 1991. ISBN 0-8186-2490-6.
5. D. W. Fife *et al.*, "Evaluation of computer-aided system design tools for SDI battle management/C3 architecture development," Inst. for Defense Analyses, Alexandria, VA, USA, IDA Paper P-2062, Oct. 1987.
6. K. Thompson, "Reflections on trusting trust," *Comm. ACM*, vol. 27, no. 8, pp. 761–763, 1984, doi: 10.1145/358198.358210.
7. N. Boucher and R. Anderson, "Trojan source: Invisible vulnerabilities," Oct. 30, 2021, doi: 10.48550/arXiv.2111.00169.
8. D. Drusinsky and J. B. Michael, "Obtaining trust in executable derivatives using crowdsourced critiques with blind signatures," *Computer*, vol. 53, no. 4, pp. 51–56, 2020, doi: 10.1109/MC.2020.2970819.
9. S. Biddle and J. Poulson, "American phone-tracking firm demo'd surveillance posers by spying on CIA and NSA," The Intercept, Apr. 22, 2022. https://theintercept.com/2022/04/22/anomaly-six-phone-tracking-zignal-surveillance-cia-nsa/ (Accessed: Apr. 28, 2022).
10. T. Brewster, "Are you one of Avast's 400 million Users? This is why it collects and sells your web habits," Forbes, Dec. 9, 2019. https://www.forbes.com/sites/thomasbrewster/2019/12/09/are-you-one-of-avasts-400-million-users-this-is-why-it-collects-and-sells-your-web-habits/?sh=47cc79392bdc (Accessed: Apr. 28, 2022).

JAMES BRET MICHAEL is a professor in the Naval Postgraduate School's Department of Computer Science and Department of Electrical and Computer Engineering, Monterey, California, 93943, USA. Contact him at bmichael@nps.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: Direct unresolved complaints to 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.

AVAILABLE INFORMATION: To check membership status, report an address change, or obtain more information on any of the following, email Customer Service at help@computer.org or call +1 714 821 8380 (international) or our toll-free number, +1 800 272 6657 (US):

- Membership applications
- Publications catalog
- Draft standards and order forms
- Technical committee list
- Technical committee application
- Chapter start-up procedures
- Student scholarship information
- Volunteer leaders/staff directory
- IEEE senior member grade application (requires 10 years practice and significant performance in five of those 10)

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 Society publishes 12 magazines, 19 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 society holds more than 215 conferences each year and sponsors many educational activities, including computing science accreditation.

Certifications: The society offers three software developer credentials.

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

MEMBERSHIP & PUBLICATION ORDERS
Phone: +1 800 272 6657; **Fax:** +1 714 816 2121;
Email: help@computer.org

EXECUTIVE COMMITTEE

President:	Jyotika Athavale
President-Elect:	Hironori Washizaki
Past President:	Nita Patel
First VP:	Grace A. Lewis
Second VP:	Nils Aschenbruck
Secretary:	Mrinal Karvir
Treasurer:	Darren Galpin
VP, Membership & Geographic Activities:	Kwabena Boateng
VP, Professional & Educational Activities:	Cyril Onwubiko
Interim VP, Publications:	Jaideep Vaidya
VP, Standards Activities:	Edward Au
VP, Technical & Conference Activities:	Terry Benzel
2023–2024 IEEE Division VIII Director:	Leila De Floriani
2024–2025 IEEE Division V Director:	Christina M. Schober
2024 IEEE Division V Director-Elect:	Cecilia Metra

BOARD OF GOVERNORS

Term Expiring 2024:
Saurabh Bagchi, Charles (Chuck) Hansen, Carlos E. Jimenez-Gomez, Daniel S. Katz, Shixia Liu, Cyril Onwubiko

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

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

IEEE BOARD OF DIRECTORS

President & CEO:	Thomas M. Coughlin
President-Elect:	Kathleen A. Kramer
Director & Secretary:	Forrest (Don) Wright
Director & Treasurer:	Gerardo Barbosa
Past President:	Saifur Rahman
Director & VP, Educational Activities:	Rabab Ward
Director & VP, Publication Services & Products:	Sergio Benedetto
Director & VP, Member & Geographic Activities:	Antonio Luque
Director & President, Standards Association:	James E. Matthews III
Director & VP, Technical Activities:	John Verboncoeur
Director & President, IEEE-USA:	Timothy T. Lee

IEEE

revised 22 February 2024

EDITOR: Tim Menzies, North Carolina State University, tim@menzies.us

DEPARTMENT: SE FOR AI

This article originally appeared in IEEE Software vol. 40, no. 2, 2023

The Engineering Mindset Is an Ethical Mindset (We Just Don't Teach It That Way... Yet)

Tim Menzies, Brittany Johnson, David L. Roberts, and Lauren Alvarez

FROM THE EDITOR

For "SE for AI" column applications, do you have a surprising result or industrial experience? Something that challenges decades of conventional thinking in software engineering? If so, e-mail a one-paragraph synopsis to tim@menzies.us (subject line: "SE for AI: Idea: [Your Idea]"). If that looks interesting, I'll ask you to submit a 1,000–3,000-word article (where each graph, table, or figure is worth 250 words) for review for *IEEE Software*. Note: Heresies are more than welcome (if supported by well-reasoned industrial experiences, case studies, or other empirical results).—*Tim Menzies*

There are far too many examples where software engineers have deployed artificial engineering (AI) models with dubious, even dangerous, ethical properties (see "On the Need for Ethical Software"). How can we fix that?

WHY TEACH MORE ETHICS?

A recent Stack Overflow survey of more than 100,000 developers[1] showed that barely half responded that they would decline to develop unethical software. In that same survey, a third of respondents said, "it depends." These signals point to a need for frameworks and tools to support developers in making ethical choices and training them to always apply those frameworks. As educators, we ask the following question: "How can we, while training future software engineers, also be training ethical decision makers?"

There are many excellent computer science (CS) subjects devoted to ethics that allow free access to all their materials. For example, in our graduate software engineering (SE) class, we get students to apply the "seven steps to ethical decision making"[2] to the numerous ethical case studies documented at onlineethics.org[3] (for more examples of this kind of excellent material, see Hill[4] and see "Other Work on Integrating Ethics Across the Computer Science Curriculum").

While we applaud the authors of that material, we are worried that ethics is often taught separate from (or as a cursory add-on to) the rest of the curriculum (often in some separate subject called "Ethics in Computer Science" or an all-too-short module within a longer course). We believe there is more value in an integrated approach where ethics is woven into every subject and material from one subject informs the ethical discussions of another. Perhaps, more substantively, we believe our profession should acknowledge, embrace, and act to ensure that an engineering mindset is an ethical mindset.

To make this concrete, at the end of this article, we offer TESTED, an example of a rewrite of a graduate automated SE course. In that rewrite, issues of ethics and responsibility are "baked into" the whole subject (and are not some clumsy post hoc add-on). TESTED is a proof by example that we can achieve the ethical engineering mindset without detracting from the core technical topics of a subject.

Digital Object Identifier 10.1109/MS.2022.3227597
Date of current version: 13 February 2023

ON THE NEED FOR ETHICAL SOFTWARE

Chapter Six of Safiya Noble's book *Algorithms of Oppression*[S1] tells the sad tale of how a design quirk of Yelp ruined a small business. As one of Noble's interviewees put it, "Black people don't 'check in' and let people know where they're at when they sit in my (hair dressing salon). They already feel like they are being hunted; they aren't going to tell the Man where they are." Hence, that salon fell in the Yelp ratings (losing customers) since its patrons rarely pressed the "checked-in" button. There are many other examples where software engineers fielded AI models, without noticing biases in those models.

- Amazon had to scrap an automated recruiting tool as it was found to be biased against women.[S2]
- A widely used face recognition software was found to be biased against dark-skinned women[S3] and dark-skinned men.[S4]
- Google Translate, the most popular translation engine in the world, shows gender bias. "She is an engineer, he is a nurse," when translated into Turkish and then again into English, becomes "He is an engineer, she is a nurse."[S5]

For our purposes, the important point of the first Noble example is this: if software designers had been more intentional about soliciting feedback from the Black community, then they could have changed how check-ins are weighted in the overall Yelp rating system. As to the other examples, in each case, there was some discriminatory effect that was easy to detect and repair,[S6] but developers just failed to test for those biases.

There is a solution to all these problems: if a small group of people builds software for the larger community, they need to listen more to the concerns of the larger community. For that to work, the smaller group of developers has to admit the larger group into their design processes—either via a) changing the reward structures such that there are inducements for the few to listen to the many (for example, by better government legislation or professional standards); b) inclusion practices that admit the broader community into the developer community; or c) review practices where the developers can take better and faster feedback from the community. To say that another way, from an ethical perspective, it is good practice to give software to someone else and let them try to break it. For a discussion on tools to discuss that process, see the TESTED toolkit (discussed at the end of this article).

REFERENCES

S1. S. U. Noble, *Algorithms of Oppression*. New York, NY, USA: New York Univ. Press, 2018.

S2. J. Dastin, "Amazon scraps secret AI recruiting tool that showed bias against women," *Reuters*, Oct. 2018. [Online]. Available: https://reut.rs/2Od9fPr

S3. L. Hardesty, "Study finds gender and skin-type bias in commercial artificial-intelligence systems," Massachusetts Inst. Technol., Cambridge, MA, USA, Feb. 2018. [Online]. Available: https://news.mit.edu/2018/study-finds-gender-skin-type-bias-artificial-intelligence-systems-0212

S4. "Wrongfully accused by an algorithm," *New York Times*, Jun. 2020. [Online]. Available: https://www.nytimes.com/2020/06/24/technology/facial-recognition-arrest.html

S5. A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, Apr. 2017. [Online]. Available: https://science.sciencemag.org/content/356/6334/183, doi: 10.1126/science.aal4230.

S6. J. Chakraborty, S. Majumder, and T. Menzies, "Bias in machine learning software: Why? How? What to do?" in *Proc. 29th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2021, pp. 429–440, doi: 10.1145/3468264.3468537.

WHO SHOULD TEACH MORE ETHICS?

Not a Problem for Software Engineers?

Some argue that the ethical issues of software are best left up to politicians (to make legislation) or lawyers (who can argue the nuances of that legislation in court). In reply, we say that the ethical problems raised by software, and in particular intelligent software, are now so complex and rapidly evolving that lawyers and politicians just cannot keep up. At least some of these ethical issues must now be measured and mitigated by the teams designing and maintaining those systems.

Further, addressing the ethical issues posed by software should not be something that engineers do out of statutory obligation but out of a sense of

OTHER WORK ON INTEGRATING ETHICS ACROSS THE COMPUTER SCIENCE CURRICULUM

We are not the only ones to discuss an integrated approach to ethics across the computer science (CS) curriculum.[S7,S8] A motivation present in the CS ethics curriculum is to contextualize technology by connecting it to its societal impact. Krakowski et al.[S9] referred to this coupling as a "sociotechnical" curriculum and highlight the curriculum's success, which corroborates with previous case studies, including a survey reviewing 115 university tech ethics course syllabi.[S7,S10] Fiesler et al.[S7] note the importance of interdisciplinary learning and the challenge to standardize teaching while considering the many variations of "tech ethics" integrations in present curricula.

Prior researchers have argued that ethics should be introduced as early as Programming 101 to prevent the "I'm just an engineer" mindset and underscored the need for more interdisciplinary collaboration with domains like philosophy to create a standard infrastructure for teaching and embedding ethics across CS. A European study[S11] on the importance of CS ethics curriculum coincides with previous research on 1) the widespread embrace for ethics curriculum integration; 2) the lack of hours dedicated to ethics teachings; and 3) the previous misconceptions of ethics referencing it as a standalone topic or area of concentration rather than a foundational concept present in main domains such as artificial intelligence, data science, and security. In accordance with the present results, studies have detailed the current "ethics crisis" and present a call for computer scientists to make the same strides as climate scientists by utilizing collaborative teaching practices and exchanges of knowledge.[S7,S12]

The suggestions for future work involve using ethics as a pedagogical lens such as "responsible" versus "irresponsible computing" frameworks[S13] and assessing academics' levels of ethical awareness to improve curriculum integration by beginning with educating professors.[S14] A challenge for future research is to expand the ethics theory at present and connect tech ethics pedagogies to feminist theory and critical inquiry. Williams et al.[S15] spotlight the dangers of minimizing ethics into "consequentialist, duty, or virtue ethics" and argue that students are not prepared for real-world scenarios involving structural societal systems of inequity.

The current state of the literature emphasizes an ethics crisis but an open community of academics willing to learn how to successfully integrate tech ethics given the right curriculum.

REFERENCES

S7. C. Fiesler, N. Garrett, and N. Beard, "What do we teach when we teach tech ethics?: A syllabi analysis," in *Proc. 51st ACM Tech. Symp. Comput. Sci. Educ.*, Feb. 2020, pp. 289–295, doi: 10.1145/3328778.3366825.

S8. B. J. Grosz et al., "Embedded EthiCS: Integrating ethics across CS education," *Commun. ACM*, vol. 62, no. 8, pp. 54–61, Aug. 2019, doi: 10.1145/3330794.

S9. A. Krakowski, E. Greenwald, T. Hurt, B. Nonnecke, and M. Cannady, "Authentic integration of ethics and AI through sociotechnical, problem-based learning," in *Proc. 12th AAAI Symp. Educ. Adv. Artif. Intell.*, Jan. 2022, pp. 12,774–12,782, doi: 10.1609/aaai.v36i11.21556.

S10. B. S. Baumer, R. L. Garcia, A. Y. Kim, K. M. Kinnaird, and M. Q. Ott, "Integrating data science ethics into an undergraduate major: A case study," *J. Statist. Data Sci. Educ.*, vol. 30, no. 1, pp. 15–28, Mar. 2022, doi: 10.1080/26939169.2022.2038041.

S11. I. Stavrakakis et al., "The teaching of computer ethics on computer science and related degree programmes. A European survey," *Int. J. Ethics Educ.*, vol. 7, no. 1, pp. 101–129, Apr. 2022, doi: 10.1007/s40889-021-00135-1.

S12. I. D. Raji, M. K. Scheuerman, and R. Amironesei, "You can't sit with us: Exclusionary pedagogy in AI ethics education," in *Proc. ACM Conf. Fairness, Accountability, Transparency*, Mar. 2021, pp. 515–525, doi: 10.1145/3442188.3445914.

S13. L. Cohen, H. Precel, H. Triedman, and K. Fisler, "A new model for weaving responsible computing into courses across the CS curriculum," in *Proc. 52nd ACM Tech. Symp. Comput. Sci. Educ.*, Mar. 2021, pp. 858–864, doi: 10.1145/3408877.3432456.

S14. R. T. Hans, S. M. Marebane, and J. Coosner, "Computing academics' perceived level of awareness and exposure to software engineering code of ethics: A case study of a South African university of technology," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 5, pp. 585–593, 2021, doi: 10.14569/IJACSA.2021.0120570.

S15. R. M. Williams, S. Smarr, D. Prioleau, and J. E. Gilbert, "Oh No, Not Another Trolley! On the need for a co-liberative consciousness in CS pedagogy," *IEEE Trans. Technol. Soc.*, vol. 3, no. 1, pp. 67–74, Mar. 2022, doi: 10.1109/TTS.2021.3084913.

obligation to our fellow human beings. We should be self-policing our practices and not outsourcing requirements to those without the technical skills of our profession. That being said, often, powerful economic, political, or social forces can (inadvertently) incentivize unethical behavior. As such, top-down governance via legislation or policy can play a role in setting the occasion to drive changes in ethical engineering practice. However, bottom-up cultural shifts can drive change more efficiently and enhance those statutory efforts (should they occur).

In our view, issues of, for example, software bias in intelligent systems are now such a pressing matter that it is reckless and irresponsible to assume some other community will fix it. This is an "all hands on deck" situation where many communities need to offer their skills to address a pressing social problem. And there is much the SE community can offer for this problem. Berk et al.[5] famously said in 2017 that "It is impossible to achieve fairness and high performance simultaneously (except in trivial cases)." But in 2021, an SE research team showed that Berk et al. were wrong since many of the things done to fix fairness are also the sampling operators widely used in software analytics to improve prediction. Hence, Chakraborty et al.[S6] showed that their Fair-SMOTE system could improve not only fairness measures but also predictive performance.

Not a Problem for Technologists?

Some say software is a technology and, as such, is selected to favor the ruling elite.[6] In this view, software is as inherently bad (racist, sexist, misinforming) as anything else selected by their social context. Hence, in that view, there is no value in fixing, for example, algorithms until we first fix the society that selects and deploys them. Noble,[S1] for example, wants "decoupling of advertising and commercial interests from the ability to access high-quality information on the Internet"; to "suspend the circulation of racist and sexist material that is used to erode our civil and human rights"; and to require that all search results be annotated to symbolize, for example, pornography (in red); business or commercial material (in green); entertainment (in orange), etc.

But is this viewpoint incomplete? Does it underestimate the number of choices within a technology (most of which are unexplored)? Not all technology inexorably returns a single output hardwired into its design. Some algorithms are exploratory tools that help humans trade off between numerous competing goals. Other software has a large set of configuration parameters that can change all manner of things, including the false positive rate between different populations. Just as algorithm designers need to know more about the broader social issues of their work, so too do social theorists need to know about algorithms. In this way, these groups can better work together while discussing, for example, how language can marginalize and disfranchise social groups[7] or how the Zitzler predicate can sort items on the Pareto frontier during multiobjective optimization.[8]

What we seek is a "two-way street" between what we might call the humanities view (which is light on CS knowledge) and the CS view (which is light on knowledge of the broader social context). For example, Gebru[9] wants regulation that "specifically says that corporations need to show that their technologies are not harmful before they deploy them." Implementing that requirement, at a company-level scale, would require many things, including the fairness testing methods discussed here.

HOW TO TEACH ETHICS (AN INTEGRATED APPROACH)

To repeat the main theme of this article: We believe it is important to integrate concepts related to ethical decision making throughout the various courses in CS curricula. More specifically, we say that every course in the CS curriculum should explore the following questions:

- Who could get hurt by the software from that subject?
- How can that hurt be mitigated? Here, students will be asked to take methods from one subject and apply them to another.
- How can that kind of software be changed to empower more people such that this kind of hurt does not happen in the future?

By answering these questions, students will be encouraged to consider who is empowered, or disempowered, by technology and the decisions they make when developing and deploying that technology. For example, students should be taught it's critical

- for someone else to be able to review and understand what you are doing
- for others to have opportunities to object to all or part of what you do
- to respond to any objections raised (perhaps by changing what you are doing).

In this view, students in, for example, a database course may be led to critically analyze how well a database supports the following:

- privacy (that is, controlling who can see what about whom)
- forgetting (that is, whether all the details about one person can be removed from the systems)
- access (that is, whether individuals can check who has seen what about their records)
- correction (for example, if a government database has some error about someone, whether someone can repair that error).

For another example, consider our TESTED data science class. For this article, the important point of TESTED is that it shows that ethical issues (specifically, issues of accountability) can augment and improve syllabus design.

TESTED is a set of coding assignments, written in Lua (which is a small and simple Python-like language but with far less overhead). Students use these samples as an executable specification that they must reproduce in any other language they like (except Lua). Each of these assignments is about one to two weeks of work. Hence, it is suitable for homework or (by combining several modules) a large end-of-term project.

Given all the examples in "On the Need for Ethical Software," testing is an essential component of AI. Hence, TESTED is very focused on test-driven development—by both developers and outside groups. TESTED assumes that the best way to test something is to give it to someone else and watch them break it. This is actually a core principle of ethical programming. Vance et al.[10] argue that a precondition for accountability is the knowledge of an external audience, who could approve or disapprove of a system. Hence, TESTED includes five accountability-support tools.

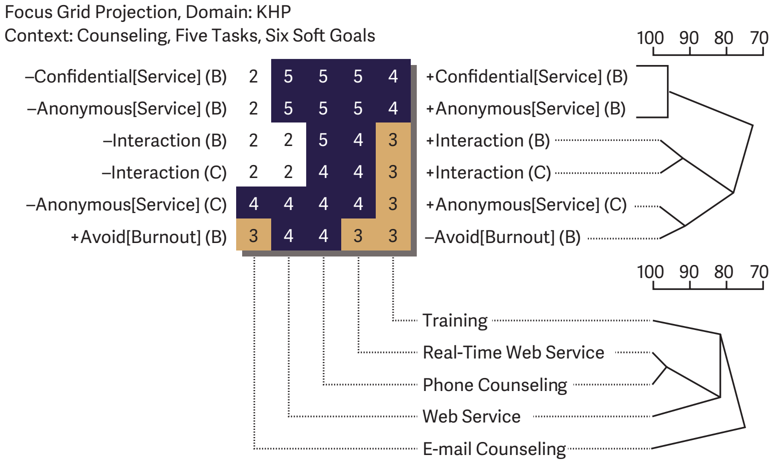
1. operators for learning the boundaries of a system's competency
2. methods for looking beyond those boundaries (taken from cognitive psychology)
3. human-readable model generation methods that can extract symbolic descriptions from training data (since that is what humans need for explaining a system)
4. cost-effective sampling methods that let outsiders probe a system, looking for interesting (or alarming) behavior
5. semisupervised learners where algorithms make conclusions based on a small sample of the total data space (so humans are not overwhelmed with excessive questions).

TESTED encourages a mix-and-match approach to AI (where developers can exert much control over what functionality they deliver). To foster that approach, students are encouraged to reflect on all the overlap within the previously mentioned accountability-support tools.

- To explore beyond the boundaries of the current system, TESTED uses hierarchical clustering to implement the tautology and instance selection hierarchies used in repertory grids (see Figure 1). Repertory grids are a tool proposed by the cognitive psychologist George Kelly as a method for eliciting tacit knowledge. Niu and Easterbrook[11] comment that repertory grids are widely recognized as a domain-independent method for externalizing individuals' personal constructs. Interviewees are invited to offer their own examples from their own domain. Then, they are asked: "Given three examples (picked at random), on what dimension is one example most different from the other two?"
- To define a system's boundary, TESTED reuses the same clustering tools applied to build the repertory grids. To check if new inputs fall within the data used to train a system, TESTED runs the new examples down its tree of clusters. New inputs are anomalous if they fall far from the median of a cluster. An anomalous example can either:
  - trigger an alert based on any conclusion for this example (since it is based on information that is out of the scope of the training set)
  - trigger model updates (and, to keep that tractable, those updates can be restricted to just the clusters suffering from anomalies).

Further to the aforementioned, there are many other ways TESTED can show that, under the hood, there is much commonality in many AI systems. For example, once we can recursively cluster data (using the methods described previously), then there are many ways to use the leaf clusters found by that recursion.

- Nearest-neighbor algorithms can be very slow. They can be sped up by looking only for near neighbors within the same leaf cluster.
- By sampling and labeling only a few examples per leaf, then our tree of clusters becomes a semisupervised learner. In this approach, all examples of a leaf share the labels seen on one (or slightly more than one) example per leaf. Semisupervised learners are very important for human-in-the-loop systems since they mean humans are pestered for their opinion on only a small subset of the data.
- By comparing the labels collected on different branches, it is possible to compute actions that adjust results from one leaf L1 to another leaf L2. Note that if the comparison is based on some multiobjective criteria, then this whole approach converts from "classification" and "regression" to a "multiobjective optimizer." Note also that by discretizing numeric ranges such that we use only ranges with different ratios of examples from L1:L2, we can then "bunch up" many ranges into far fewer ranges (which makes rule or decision tree generation much simpler and faster).
- Once the leaves are sorted, then those actions can be reported to the members as either
  - plans on how to improve things (that is, how to change a conclusion from a worse leaf to a better leaf)
  - monitors of what conditions can change conclusions from a better to a worse leaf.


Focus Grid Projection, Domain: KHP
Context: Counseling, Five Tasks, Six Soft Goals
100 90 80 70
100 90 80 70

	E-mail Counseling	Web Service	Phone Counseling	Real-Time Web Service	Training	
–Confidential[Service] (B)	2	5	5	5	4	+Confidential[Service] (B)
–Anonymous[Service] (B)	2	5	5	5	4	+Anonymous[Service] (B)
–Interaction (B)	2	2	5	4	3	+Interaction (B)
–Interaction (C)	2	2	4	4	3	+Interaction (C)
–Anonymous[Service] (C)	4	4	4	4	3	+Anonymous[Service] (C)
+Avoid[Burnout] (B)	3	4	4	3	3	–Avoid[Burnout] (B)

FIGURE 1. An intro TESTED homework: implement the row/column clustering of a repertory grid. (Source Niu and Easterbrook.[11])

Returning now to the issue of ethics and accountability (and the need for outsiders to test a system), the previously mentioned description of TESTED simplifies the interactions between human and AI systems. Semisupervised learners reduce the number of interactions required between humans and AI. Visualizations such as Figure 1 offer high-level symbolic descriptions needed for human comprehension (and for a more detailed view, we can use decision trees). Anomaly detectors tell us when a current model needs to be extended. Repertory grids let us explore the space of concepts important to users. Since our RepGrid analysis scales from small examples (like Figure 1) to much larger datasets, we can run the same attribute/example clustering on the concepts of the users.

THE ENGINEERING MINDSET CAN BE AN ETHICAL MINDSET

> Our challenge in building a professional culture comprising an ethical mindset is not to appropriate the success of technical fields but to replicate and adapt for our purposes. Technology rarely, if ever, exists for technology's sake. The artifacts engineers produce exist in a context comprising the environment and people that interact with the artifact or are impacted by it. These artifacts aren't purely technical, they're sociotechnical, and traditional engineering education often eschews the socio in favor of the technical.

In this short article, we have offered motivation and an example of why and how ethics can be an integral part of contemporary CS curricula. Bringing the socio and technical closer to parity of focus in engineering education doesn't represent a monumental change in the skills we teach; it simply requires that ethical considerations be taught to be as natural a part of the solution to an engineering problem as the choice of compiler, programming language, dataset, etc. in SE.

The tools of other communities may provide inspiration, but we as a community of engineering professionals already have a well-stocked toolbox from which we can build a broad solution to training ethically minded future engineers. The engineering mindset is often described as comprising, in large part, "systems thinking" or the idea that the world around us is linked in (often) subtle but critical ways. Systems thinking promotes maintaining awareness of these relationships to identify structures and to be conscious of how choices interact with those structures. In essence, the engineering mindset can be thought of as a dedication to asking the right questions, at the right times, and taking into account the right relationships. This is not at all different from the ethical mindset, where asking the right questions, at the right time, and accounting for the right people are fundamental.

In short, engineers trained to have an engineering mindset already have the skills to acquire an ethical mindset—they need the context, motivation, and experience to engage in these systems thinking tasks, keeping the people who are a part of the structure of these systems on par with the technical. Ethical frameworks, like TESTED, can be a powerful tool in instilling broad adoption of the ethical mindset when deployed widely in education.

REFERENCES

1. "Developer survey results 2018." Stack Overflow. Accessed: Dec. 1, 2022. [Online]. Available: https://insights.stackoverflow.com/survey/2018
2. "Geoethics: Ethical decision-making." SERC. Accessed: Dec. 1, 2022. [Online]. Available: https://serc.carleton.edu/geoethics/Decision-Making
3. "Resources. Online Ethics Center." Accessed: Dec. 1, 2022. [Online]. Available: https://onlineethics.org/resources?combine=software&field_keywords_target_id=&field_resource_type_target_id=13236
4. E. M. Peck. "Integrating social responsibility into core CS." GitHub. Accessed: Dec. 1, 2022. [Online]. Available: https://evanpeck.github.io/projects/responsibleCS
5. R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth, "Fairness in criminal justice risk assessments: The state of the art," 2017, *arXiv:1703.09207v1*.
6. D. F. Noble, *America by Design*. New York, NY, USA: Knopf, 1977.
7. P. Mpofu and A. Salawu, "Linguistic disenfranchisement, minority resistance and language revitalisation: The contributions of ethnolinguistic online communities in Zimbabwe," *Cogent Arts Humanities*, vol. 5, no. 1, Nov. 2018, Art. no. 1551764, doi: 10.1080/23311983.2018.1551764.
8. E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Int. Conf. Parallel Probl. Solving Nature*, Springer-Verlag, 2004, pp. 832–842, doi: 10.1007/978-3-540-30217-9_84.
9. K. Adams. "Timnit Gebru envisions a future for smart, ethical AI." Marketplace. Accessed: Dec. 1, 2022. [Online]. Available: https://www.marketplace.org/shows/marketplace-tech/timnit-gebru-envisions-a-future-for-smart-ethical-ai/
10. A. Vance, P. B. Lowry, and D. Eggett, "Increasing accountability through user-interface design artifacts: A new approach to addressing the problem of access-policy violations," *MIS Quart.*, vol. 39, no. 2, pp. 345–366, Jun. 2015, doi: 10.25300/MISQ/2015/39.2.04.
11. N. Niu and S. Easterbrook, "So, you think you know others' goals? A repertory grid study," *IEEE Softw.*, vol. 24, no. 2, pp. 53–61, Mar./Apr. 2007, doi: 10.1109/MS.2007.52.

TIM MENZIES is a full professor at North Carolina State University, Raleigh, NC 27606 USA. Contact him at timm@ieee.org.

BRITTANY JOHNSON is an assistant professor at George Mason University, Fairfax, VA 22030 USA. Contact her at johnsonb@gmu.edu.

DAVID L. ROBERTS is an associate professor at North Carolina State University, Raleigh, NC 27606 USA. Contact him at dlrober4@ncsu.edu.

LAUREN ALVAREZ is a Ph.D. candidate at North Carolina State University, Raleigh, NC 27606 USA. Contact her at lalvare@ncsu.edu.

EDITORS: Munindar P. Singh, mpsingh@ncsu.edu,
Pradeep K. Murukannaiah, P.K.Murukannaiah@tudelft.nl

DEPARTMENT: INTERNET ETHICS

This article originally appeared in
IEEE Internet Computing
vol. 27, no.2, 2023

Toward an Ethical Framework for Smart Cities and the Internet of Things

Munindar P. Singh, *North Carolina State University, Raleigh, NC, 27695, USA*

Pradeep K. Murukannaiah, *Delft University of Technology, 2628, Delft, The Netherlands*

As smart cities increasingly become real, an ethical framework for them becomes increasingly necessary. Surprisingly, current approaches largely disregard such a framework and concentrate primarily on challenges pertaining to the data lifecycle. However, a smart city involves much more than data gathering: it involves the interactions of residents, businesses, and government agencies with respect to public and private resources subject to potentially subtle regulations and other norms. This article introduces a sociotechnical view of smart cities and shows how it may be profitably mapped to the moral foundation theory to provide a comprehensive ethical framework.

A *smart* city is one that involves the digitalization of its infrastructure and services to meet goals, such as improving residents' well-being and realizing gains in terms of efficiency, cost, sustainability, and resilience to natural and societal disruptions. Definitions by the Organization for Economic Co-operation and Development (OECD, an international body)[1] and the National Institute of Standards and Technology (NIST, a US body)[2] agree on this core idea though they may differ in some details and phrasing.

The vision of smart cities unites progress in sensor technologies, storage, communications, information processing architectures, and data analytics to build an Internet of Things (IoT) system at city scale. This vision is attractive because it brings together complementary technologies to generate innovative solutions that can help ordinary people.

Smart cities are motivated by several potential goals, including the following extracted from a recent report by NIST.[2]

1) Faster and wider delivery of urban services.
2) A reduction in costs of operating a resident-responsive infrastructure.
3) Increased opportunities for interaction, collaboration, and commerce between residents, businesses, and government agencies.
4) Enhanced environmental sustainability.
5) Support for equitable access to city services and related services, such as healthcare.
6) Improved quality of life for residents.

Important themes in smart cities include transportation[3] and the smart grid,[4] although the applications include virtually all aspects of civic life, including healthcare, education, space usage, and waste management.[a]

Smart cities and IoT

We conceive of a smart city as an agglomeration of IoT applications with a human and societal flavor. Indeed, when we think of the IoT in general, besides the purely technical development—such as the miniaturization of sensors, power packs, and radios—the major concerns of the IoT are reflected in smart cities. This is because smart cities bring together heterogeneity, multiple stakeholders and administrative domains, and continual negotiation of functional and nonfunctional requirements of multiple interacting

1089-7801 © 2023 IEEE
Digital Object Identifier 10.1109/MIC.2023.3236826
Date of current version 14 March 2023.

[a] [Online]. Available: https://www.asme.org/topics-resources/content/top-10-growing-smart-cities

IoT applications. For this reason, we see smart cities as a challenging exemplar of an IoT application. Although we focus on an ethical framework for smart cities, we see this framework as applying to even narrower IoT settings, such as individual applications.

Scope of this article

This article outlines the key elements of a comprehensive ethical framework for smart cities. To this end, it adopts a sociotechnical, yet computational stance on smart cities. Ethics can refer to a variety of concerns. This article adopts a viewpoint based on the well-known moral foundations theory (MFT).[5] This theory is well-suited to smart cities because it encompasses the key moral dimensions or *foundations* that pertain to interactions in a city: impacts on the interests of individuals (e.g., residents and businesses) and institutions, statistical properties of gains and losses, imposition of power on individuals, and protection of residents' values. In this way, MFT goes beyond a focus on data privacy to the essence of the lives of the people who form a city. Further, this article maps interactions in a smart city viewed as a sociotechnical system to the moral foundations.

TRADITIONAL THINKING AND ITS LIMITATIONS

The first generation of smart cities is organized around well-defined existing services and is realized mainly through technology upgrades for sensing, communications, and computing. For example, the US Department of Transportation[3] identifies efforts on creating plans for improving street lighting, measuring congestion and improving transportation throughput on road networks, measuring air pollution, and improving mobility for residents.

OECD[1] motivates the fact that the introduction of smart cities may lead to challenges, such as privacy and widening inequality as well as challenges in regulation, broadly concerning government contracts and labor laws.

Current discussions of smart cities often involve little more than inserting the word "smart" before virtually anything that one may associate with a city. As a case in point, a recent NIST publication,[2] [Figure 3, p. 6] lists smart education, smart resource and waste management, smart communications, smart health, smart urban planning, smart building, smart grid, smart security, smart mobility, smart environment, and (quite mysteriously) smart citizens. The repeated use of "smart" as a buzzword does little to lend clarity to the conception. Even without the buzzword, the conception is very much based on the services available in traditional cities.[b] Consequently, a major concrete shortcoming of current thinking is that it either involves interactions between a resident and a government agency or leaves the interactions unspecified and amorphous.

Despite the promise of smart cities and the recent progress in deploying IoT technologies, the current thinking on these topics focuses heavily on the technological element. That is, there is ample discussion of sensors and technological challenges, such as powering the sensors and processing the data streams they produce. However, current research is limited with respect to the human and social elements.

Some researchers mention the importance of human and social elements and acknowledge the need for a sociotechnical approach to understanding smart cities. However, when researchers talk of social aspects, they largely mean either: 1) the economic aspects, such as the relationship between the adoption of smart city technologies and pricing and incentives, or 2) the concerns inherent in the collection of data from smart infrastructure.

Specifically, current scholarship largely eschews discussions of ethics and safety in a computational manner. The closest studies address privacy and cybersecurity.[6] We found only one substantive study on ethics in smart cities[7] and even that is focused on big data and privacy. The study identifies transportation as the main illustration and considers only the aspects of transportation by itself (i.e., the data obtained through it, and the potential benefits and harms). It thus exemplifies another shortcoming of current thinking, which is to view smart city applications in a siloed manner.

TOWARD A COMPREHENSIVE ETHICAL FRAMEWORK

In contrast to the previous approaches, we posit that an ethical framework for smart cities must be comprehensive in three respects so that smart cities achieve the positive vision that computer scientists and laypeople have for them. First, it must tackle the synthesis of multiple applications. Second, whereas privacy is indeed important, the framework must address broader ethical challenges. Third, the ethical framework must accommodate relevant interactions between different types of stakeholders.

Figure 1 shows a schematic of the ethical framework we envision for smart cities.

[b] [Online]. Available: https://www.visualcapitalist.com/anatomy-smart-city/

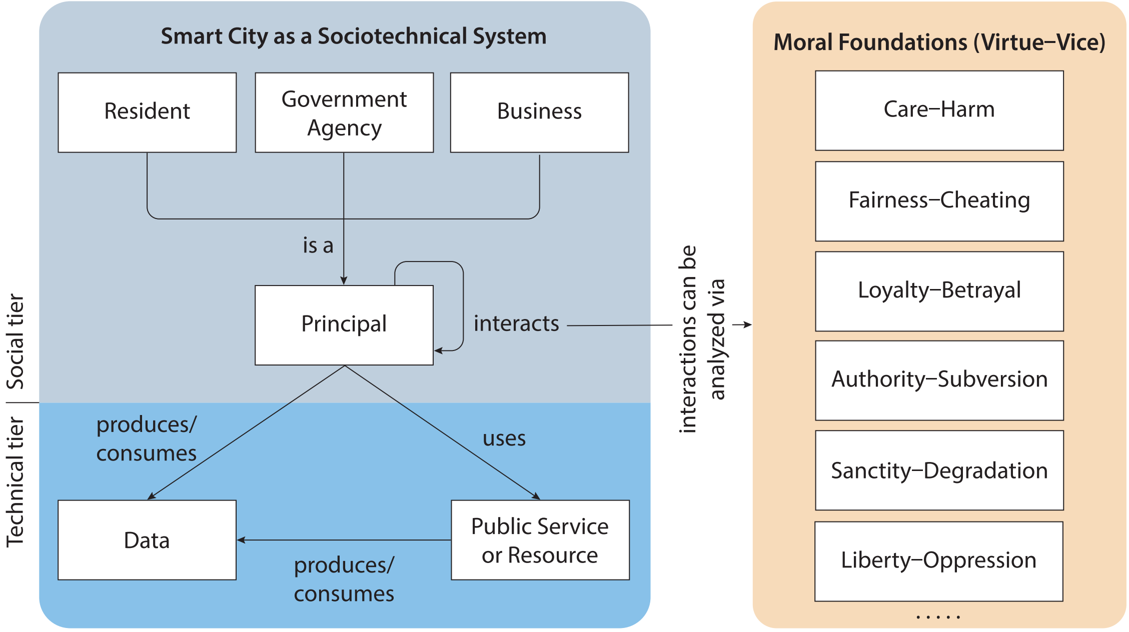


FIGURE 1. We view a smart city as a sociotechnical system, where principals interact, use public services and resources (e.g., transportation), and produce or consume data. Addressing the ethical challenges of a smart city requires shifting the understanding of ethics from the technical tier (involving data and resources) to the social tier where principals interact.

Smart City as a Sociotechnical System

A key feature of our framework is the representation of a smart city as a sociotechnical system.[8] This representation broadens the focus of ethics from a purely technical perspective (e.g., statistically mitigating biases in data) to a sociotechnical perspective, where ethics is about understanding one principal's concern for another as observed in the (technology-mediated) interactions among the principals (e.g., tracing biases in data to the data originators).[9]

The left part of Figure 1 shows the key components of a sociotechnical system in a schematic form. A principal is a social actor, such as an individual (e.g., a resident), a private institution (e.g., a business), or a public institution (e.g., a transportation agency). The principals use the services and resources available in the system and interact with each other in that process. The purpose of technology is to mediate the interactions among the principals to yield a high quality of service (e.g., in terms of efficiency, accessibility, flexibility, and so on).

Moral Foundations

We seek to analyze whether the technology-mediated interactions in a sociotechnical system yield ethical outcomes or not. To do so, we adopt a theory of moral constructs available in the literature.

The MFT advocates that "morality is about how individuals ought to relate to, protect, and respect other individuals," in line with several other moral philosophers.[5] MFT subscribes to the idea of moral pluralism, i.e., ascribing an individual's morality to a combination of multiple *moral foundations*. Thus, the moral foundations provide a language in which to describe moral behavior.

MFT includes the following six moral foundations, where each foundation is represented as a dimension ranging between a virtue and a vice (also shown on the right-hand side of Figure 1).

Care–Harm is about protecting the vulnerable, especially one's kin.

Fairness–Cheating is about ensuring that others are treated fairly.

Loyalty–Betrayal is about acting in the interest of one's social group.

Authority–Subversion is about respecting hierarchies to ensure obedience and deference toward authoritative institutions, such as courts.

Sanctity–Degradation is about disgust toward contamination.

Liberty–Oppression is about resisting domination, including working in solidarity to oppose oppression.

Illustrating the Framework

The foregoing schematic identifies the key types of principals in a smart city. In the following, we discuss the kinds of ethical challenges that arise in a smart city.

As a running example, consider that public transportation is the main means of travel in a smart city. The transportation agency of the city collects data about the occupancy of the vehicles, including the number of passengers and how they traveled (e.g., first-class versus economy) in each vehicle. For transparency, the agency shares these data publicly.

Although the transparency is desirable, as we show in the following, many ethical challenges arise in how the data are used when the transportation service is combined with other public services and business activities occurring in a city. To this end, imagine that this city has a public park that has been endowed with sensors (including cameras) and actuators (such as gates). In the social tier, we would see norms (whether regulations or informal), such as using the park only within daylight hours and avoiding a picnic area where a family is already enjoying a meal.

Residents interacting with government agencies: This is the setting most discussed in the smart cities literature. When residents use a tram or enter a park, they interact with the respective municipal agencies.

As existing approaches indicate, such interactions raise the expected privacy concerns of resident data being gathered, stored, and used. In addition, the ubiquity of smart city services can lead to fine-grained tracking and control of residents. For example, the government can combine information on when someone boards or alights from a tram and when they enter and exit a park to determine their entire schedule and who they likely interact with.

The concern highlighted by these examples is consent for actions pertaining to a resident's information. Consent maps to the Care–Harm foundation because obtaining consent legitimately[10] is a way of protecting a resident's interests and values.

Residents interacting with public resources: A more important situation is when residents who use a public resource want to specify the applicable norms so they can share—more broadly, govern[8]—the resource effectively. For example, residents would need to agree on whether cameras as installed in a park (technical tier) and norms about when recording is turned ON, whether they are live monitored without recording, and how the recordings are used (social tier).

With live monitoring, cameras could enable crime prevention as well as apps to avoid congestion (by letting people know if there was space). With recording, cameras could enable solving crime. Besides crimes, behaviors, such as littering or public urination that make a space unusable for others, are common challenges with public infrastructures. Cameras can serve as deterrents and enable better routing of cleaning crews when needed. Different residents may have competing values and preferences. For example, monitoring and recording would promote safety and may be desirable to parents of young children but demote privacy, which may be dear to others. Even for people who have no intention of acting wrongly, the possibility of being observed can have a chilling effect on their behavior.

This concern maps to the Authority–Subversion foundation because of the need for residents to establish a legitimate authority (in the nature of the norms) and avoid subverting it. The specific concern of exposure resulting in a chilling effect maps to the Liberty–Oppression foundation. The concern about cleanliness maps to the Sanctity–Degradation foundation.

Residents interacting with residents: Smart cities enable residents to share resources belonging to one another or directly engage with each other in using public services. The technology in a smart city can enhance the quality of such interactions among the residents. For instance, in our example, since the transportation agency shares occupancy data, residents can choose to travel during less-crowded hours (indicating sharing of a public resource) and have conversations with fellow passengers (indicating peer-to-peer interactions). Moreover, the smart city may enable residents to carpool from their homes to and from a tram station and possibly chaperone each other's children to dance practice.

However, the technology can also be misused. For example, pickpockets in the city can use the occupancy data to target crowded vehicles, especially those where riders are chaperoning multiple children. Thus, new interactions enabled by the smart city may lead to both care and harm. Thinking further about this problem based on our ethical framework, a potential solution is to increase security in public transportation during crowded hours, which relates to the Authority–Subversion foundation.

Residents interacting with businesses: A smart city would enable improvements in efficiencies in interactions between businesses and people. Suppose a cafe is placed not far from the city park and a tram stop. Knowing when the next few trams are expected and how full they are, the cafe can plan to have coffee and tea ready and hot buns in the oven in time for the expected influx of customers. This simple optimization would reduce congestion at the cafe and reduce the time to serve customers, benefiting both the cafe's operations and the customers.

The cafe can improve customer experience further if it obtains additional information about the incoming tram riders: their ages and economic statuses and how many are in first-class or regular carriages (even without knowing their identities). But ethical hazards lurk here. The cafe may want to build a customer base focused on rich customers, who buy expensive drinks.

Suppose the cafe uses the transportation data to focus on prepping products for these customers. Thus, rich customers benefit (in terms of wait time and quality of experience) and the cafe benefits (in terms of revenue) from the smart city technology but ordinary residents suffer through increased delay and the discomfort and risk of staying in a congested store longer than otherwise. In other words, the technology would facilitate unethical (or unlawful) discrimination.

The equity concern raised in this example primarily maps to the Fairness–Cheating foundation with the risk (e.g., of exposure to infection) due to congestion secondarily mapping to the Care–Harm foundation.

Businesses interacting with government agencies: These ethical concerns in these interactions resemble those between residents and government agencies in that the government can harm or unfairly treat a small business. However, larger businesses, e.g., those that control valuable real estate, may control governmental decisions to the detriment of residents and small businesses. For example, they could leverage their power to locate tram stations conveniently for their customers as opposed to others—a strike on the Fairness–Cheating foundation. Or, by sporadically donating a large volume of groceries to a food bank (indicative of Care), they could discourage small scale but sustained donors by causing their small donations to be wasted (causing them to feel betrayal after their difficult, albeit meager efforts).

Toward a Framework

As the foregoing examples show, an ethical framework for smart cities must accommodate the moral aspects of interactions between residents, viewed (in a sociotechnical light) in conjunction with sensors and data technologies. To realize a smart city ethically is not merely to deploy the technologies or even launch individual applications but to reflect on their intended and unintended interactions with human behavior and the ramifications of those interactions on the moral foundations that motivate humans. The framework would be instantiated in methodologies for the design, deployment, and continual maintenance and re-engineering of smart city services. These methodologies would evaluate these services individually and in combination through the lens of the interactions they support between stakeholders, evaluating their outcomes on the relevant moral foundations. These methodologies and the services they produce would respect everyone's autonomy and facilitate innovative uses, and continually incorporate creative ideas.[11]

DISCUSSION

The set of moral foundations is not closed and may be extended as additional evidence or understanding of their existence arises.[5] However, the current version is adequate to show the richness of the moral realm that an ethical framework for smart cities and IoT ought to address, not merely privacy.

Approaches focused on local governance in smart cities are well-aligned with our framework. Razaghi and Finger[12] address the limitations of current reductionist approaches to smart cities and propose a sociotechnical approach that would respect residents' autonomy. Their scope differs from ours in that they focus on public administration (including municipal politics). However, their notion of sociotechnical systems is conventional and lacks a computational model. Almeida et al.[13] as well bring out the need for transparency and control.

Kontokosta and Hong[14] highlight how resident–government interactions can suffer from a lack of equity and fairness arising from how data are collected and used. Although their focus is on city services, their discussion of ethical concerns besides privacy is compatible with our framework.

Serrano et al.[2] provide a framework for key performance indicators (KPIs) for smart cities that includes selection and prioritization of city goals to enable their quantification into KPIs based on the available data. Serrano et al. recognize that different members (e.g., communities) of a city may have different priorities as regards the goals and KPIs. Their metrics address the alignment of KPIs with the priorities of the members of a city in terms, e.g., of investments made in a city. However, these metrics are somewhat *ad hoc* and rely on people producing numbers based on intuition. Still, this framework could be enhanced to produce KPIs for the relevant ethical challenges as they are mapped to various foundations.

ACKNOWLEDGMENTS

MPS thanks the NSF under Grant IIS-2116751 for partial support for this research.

REFERENCES

1. OECD, "Measuring smart cities' performance: Do smart cities benefit everyone ?," Dec. 2020. Organisation for Economic Co-operation and Development. Accessed: Dec. 27, 2022. [Online]. Available: https://www.oecd.org/cfe/cities/Smart-cities-measurement-framework-scoping.pdf
2. M. Serrano et al., "Smart cities and communities: A key performance indicators framework," NIST Special Publication 1900-206, National Institute of Standards and Technology, Gaithersburg, MD, USA, Feb. 2022, doi: 10.6028/NIST.SP.1900-206-upd1.
3. DoT, "Smart city challenge: Lessons for building cities of the future," Dec. 2016. Accessed: Dec. 27, 2022. [Online]. Available: https://www.transportation.gov/policy-initiatives/smartcity/smart-city-challenge-lessons-building-cities-future
4. Antonio Gómez Expósito et al., "City-friendly smart network technologies and infrastructures: The Spanish experience," *Proc. IEEE*, vol. 106, no. 4, pp. 626–660, Apr. 2018, doi: 10.1109/JPROC.2018.2793461.
5. J. Graham et al., "Moral foundations theory: The pragmatic validity of moral pluralism," in *Advances in Experimental Social Psychology*, P. Devine and A. Plant, Eds, vol. 47. Cambridge, MA, USA: Academic Press, 2013 pp. 55–130, doi: 10.1016/B978-0-12-407236-7.00002-4.
6. E. Ismagilova, D. L. Hughes, N. P. Rana, and Y. K. Dwivedi, "Security, privacy and risks within smart cities: Literature review and development of a smart city interaction framework," *Inf. Syst. Front.*, vol. 24, no. 2, pp. 393–414, Apr. 2022, doi: 10.1007/s10796-020-10044-1.
7. V. Chang, "An ethical framework for big data and smart cities," *Technological Forecasting Social Change*, vol. 165, pp. 120559:1–120559:11, Apr. 2021, doi: 10.1016/j.techfore.2020.120559.
8. M. P. Singh, "Norms as a basis for governing sociotechnical systems," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, pp. 21:1–21:23, Dec. 2013, doi: 10.1145/2542182.2542203.
9. P. K. Murukannaiah, N. Ajmeri, C. M. Jonker, and M. P. Singh, "New foundations of ethical multiagent systems," in *Proc. 19th Int. Conf. Auton. Agents MultiAgent Syst.*, 2020, pp. 1706–1710, doi: 10.5555/3398761.3398958.
10. M. P. Singh, "Consent as a foundation for responsible autonomy," in *Proc. 36th AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 11, pp. 12301–12306, doi: 10.1609/aaai.v36i11.21494.
11. P. K. Murukannaiah, N. Ajmeri, and M. P. Singh, "Acquiring creative requirements from the crowd: Understanding the influences of personality and creative potential in crowd RE," in *Proc. IEEE 24th Int. Requirements Eng. Conf.*, 2016, pp. 176–185, doi: 10.1109/RE.2016.68.
12. M. Razaghi and M. Finger, "Smart governance for smart cities," *Proc. IEEE*, vol. 106, no. 4, pp. 680–689, Apr. 2018, doi: 10.1109/JPROC.2018.2807784.
13. V. A. F. Almeida, D. Doneda, and E. M. da Costa, "Humane smart cities: The need for governance," *IEEE Internet Comput.*, vol. 22, no. 2, pp. 91–95, Mar. 2018, doi: 10.1109/MIC.2018.022021671.
14. C. E. Kontokosta and B. Hong, "Bias in smart city governance: How socio-spatial disparities in 311 complaint behavior impact the fairness of data-driven decisions," *Sustain. Cities Soc.*, vol. 64, 2021, Art. no. 102503, doi: 10.1016/j.scs.2020.102503.

MUNINDAR P. SINGH is a professor in computer science and a co-director of the Science of Security Lablet at NC State University, Raleigh, NC, 27695, USA. His research interests include the engineering and governance of sociotechnical systems, and AI ethics. Singh received his Ph.D. degree in computer sciences from The University of Texas at Austin, Austin, TX, USA. He is a fellow of AAAI, AAAS, ACM, and IEEE. Contact him at singh@ncsu.edu.

PRADEEP K. MURUKANNAIAH is an assistant professor in the Interactive Intelligence group at TU Delft, 2628, Delft, The Netherlands. His research focuses on engineering socially intelligent applications. Murukannaiah received his Ph.D. degree in computer science from NC State University, Raleigh, NC, USA. Contact him at p.k.murukannaiah@tudelft.nl.

DEPARTMENT: EXPERT OPINION

This article originally appeared in IEEE micro vol. 42, no. 6, 2022

RadioML Meets FINN: Enabling Future RF Applications With FPGA Streaming Architectures

Felix Jentzsch, *Paderborn University, 33098, Paderborn, Germany*

Yaman Umuroglu, Alessandro Pappalardo, and Michaela Blott, *AMD, D24 T683, Dublin, Ireland*

Marco Platzner, *Paderborn University, 33098, Paderborn, Germany*

Deep neural networks (DNNs) are penetrating into a broad spectrum of applications and replacing manual algorithmic implementations, including the radio frequency communications domain with classical signal processing algorithms. However, the high throughput (gigasamples per second) and low latency requirements of this application domain pose a significant hurdle for adopting computationally demanding DNNs. In this article, we explore highly specialized DNN inference accelerator approaches on field-programmable gate arrays (FPGAs) for RadioML modulation classification. Using an automated end-to-end flow for the generation of the FPGA solution, we can easily explore a spectrum of solutions that optimize for different design targets, including accuracy, power efficiency, resources, throughput, and latency. By leveraging reduced precision arithmetic and customized streaming dataflow, we demonstrate a solution that meets the application requirements and outperforms alternative FPGA efforts by 3.5× in terms of throughput. Against modern embedded graphics processing units (GPUs), we measure >10× higher throughput and >100× lower latency under comparable accuracy and power envelopes.

Deep learning is rapidly expanding into new horizons, including the communications space. Traditionally, this is a mature field of engineering, where solutions are carefully crafted by experts. However, the ever-increasing complexity of communication networks has prompted the exploration of deep neural networks (DNNs) for various use cases, ranging from traffic monitoring tasks to the physical interface design of radio frequency (RF) systems.[1] The latter is one example of the "RadioML" domain, where conventional radio signal processing is replaced by DNN-based processing. While this approach has shown great potential,[2] it also comes with great challenges, as radio signals are handled exclusively at the edge, often on highly constrained mobile devices that lack the compute or energy budget to run modern DNNs with sufficient performance. Traditional compute accelerators, such as graphics processing units (GPUs), are well-optimized for the huge DNNs in vision-based applications, but models used for RadioML pose a different challenge as they are typically much smaller and operate on short frames of time-series data. In turn, the live processing of an RF signal demands extreme throughput and ultra-low latency, which lies well beyond what is currently possible with GPUs.

To tackle these unique challenges of RadioML, joint specialization of the hardware accelerator and the DNN itself is key. Field-programmable gate arrays (FPGAs) offer the flexibility and parallelism to satisfy these requirements, but harnessing this potential is difficult, especially for nonexperts or under tight development time constraints. Here, we make the case for automatically-generated, custom-tailored accelerators for quantized DNNs, enabled by Xilinx' open-source FINN compiler framework. These accelerators follow the *streaming dataflow* architectural paradigm, involving layer-parallel processing of

0272-1732 © 2022 IEEE
Digital Object Identifier 10.1109/MM.2022.3202091
Date of current version 28 October 2022.

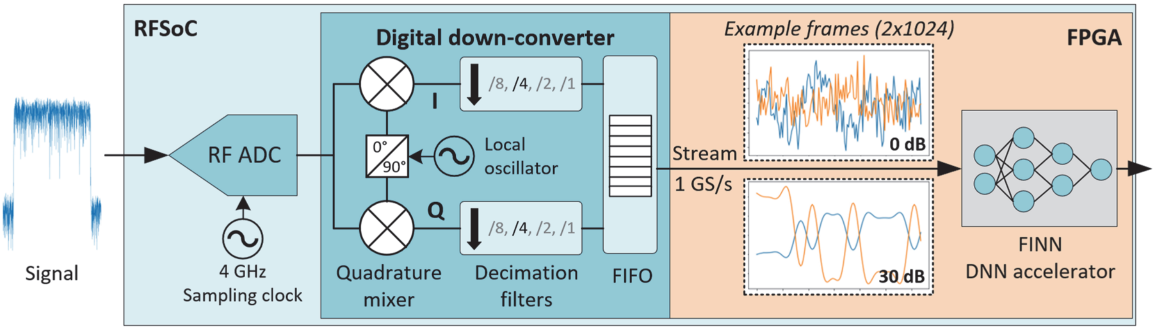


FIGURE 1. RFSoC integration of (simplified) RF front-end and DNN accelerator. Input frame examples are taken from the modulation classification dataset and come in the form of baseband in-phase/quadrature (I/Q) components (orange/blue).

the input stream, which departs from conventional compute arrays and enables deep pipelining and memory access minimization.

By way of example, we study one of the most popular RadioML use cases: automatic modulation classification. In this task, a DNN is trained to classify the modulation scheme [e.g., frequency modulation (FM), binary phase-shift keying (BPSK), quadrature amplitude modulation (QAM)-16, etc.] of a received signal. We use the open "RadioML 2018" dataset from DeepSig,[3] which covers a wide range of modulation types and signal-to-noise ratios (SNRs) and provides a baseline 1-D convolutional neural network (CNN), showing that it can outscore traditional classification based on engineered features, such as higher order moments.

In this work, we first make the case for a well-suited target platform for modulation classification and present our end-to-end tool flow to accelerate the involved CNN. Then, we report on several FINN-generated prototypes and compare them with related FPGA implementations and GPU platforms.

CASE FOR RadioML ON RFSoC

Next-generation radio architectures need platforms that can address a wide range of requirements with the same basic hardware. This adaptability is critical to accommodate emerging and ever-changing standards. FPGAs have historically provided flexible solutions for implementing the digital front-end and interfacing requirements of recent radio generations. RF system-on-chip (RFSoC) devices, such as Xilinx' Zynq UltraScale RFSoC, improve this level of flexibility by integrating RF-sampling data converters into a single chip, next to an ARM-based processing system and programmable logic fabric. Direct RF sampling, together with optimized digital signal processing (DSP) engines, offers a much more flexible approach to traditional analog frequency translation and filtering by enabling much of the signal processing to be done in the digital domain. This also eliminates the need for external input/output interfaces, which can consume a significant amount of power. In addition, the availability of FPGA programmable logic on the same device enables direct integration of downstream applications, such as DNN processing.

Figure 1 shows how such a DNN accelerator can be integrated on an RFSoC for classifying the modulation of a received I/Q modulated signal, commonly used for software-defined radios (SDRs). The datapath begins with the analog-to-digital converter (ADC), which samples the RF signal at up to 4 gigasamples per second (GS/s). Next, the samples pass through the configurable digital down-converter block, where a quadrature mixer shifts the signal from its carrier frequency down to the equivalent baseband signal. The resulting streams of in-phase (I) and quadrature (Q) components can then be down-sampled by a factor of one to eight using decimation filters before they are stored in a first-in–first-out (FIFO) gearbox buffer, resulting in an output data rate between 4 and 0.5 GS/s for the maximum ADC sample rate. In the case of our modulation classification example, this I/Q signal representation corresponds to approximately 2.5 million frames of length 1,024. Figure 1 illustrates two exemplary frame segments of BPSK-modulated signals at 0- and 30-dB SNR. Data streams within the FPGA are implemented via the AXI-Stream protocol and feed in and out of the inference accelerator. The final output is a frame's classification result.

FINN FRAMEWORK FOR STREAMING DNN ARCHITECTURES

The open-source framework FINN[4] generates specialized DNN accelerators for FPGAs using streaming dataflow architectures, with the hardware architecture customized to the specifics of a DNN topology

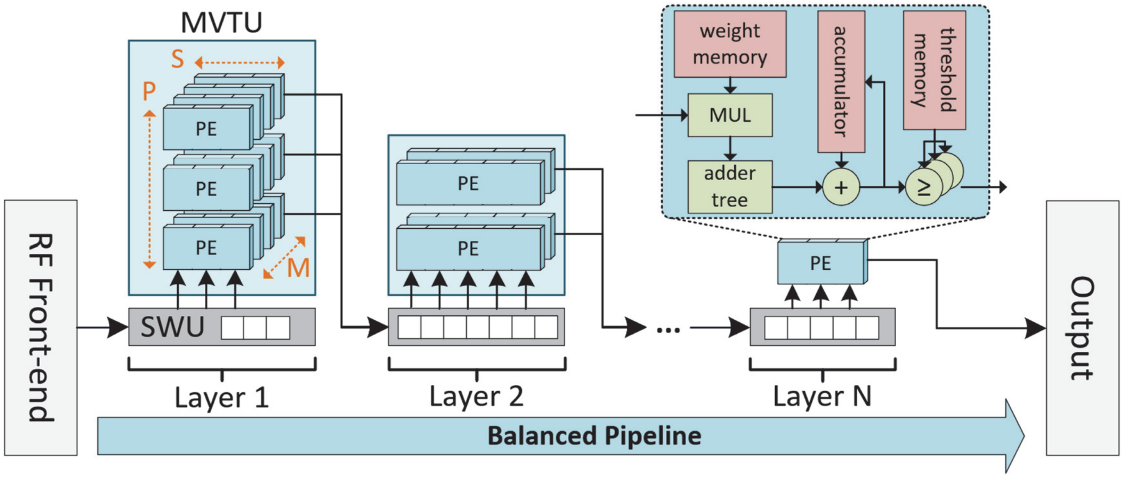


FIGURE 2. Simplified architectural overview of the FINN-generated streaming accelerator. Omits padding, pooling, and gearbox buffers between layers.

and particular datatypes used. Each layer is instantiated with its designated compute units in hardware. On-chip data streams interconnect the compute units to form the desired network topology. The small and compact size of reduced-precision quantized DNNs (QNNs) allows us to scale performance of the accelerator via reduced resource requirements of lower precision operators and store all parameters on the chip, thus avoiding external memory bottlenecks. To achieve high accuracy for low precision QNNs, we leverage the open-source PyTorch library Brevitas.[5]

Hardware Architecture

Figure 2 visualizes the FINN-generated architecture for a CNN, which comprises a balanced pipeline of computing blocks connected through on-chip streams. FINN maps each fully connected DNN layer to a dedicated compute block, the matrix–vector threshold unit (MVTU). The MVTU performs matrix multiplication between input activations and weights, followed by the so-called "multithreshold" operation, which applies the nonlinear activation function and quantization to the output in a single, efficient step. Convolution layers are based on the same MVTU structure by lowering them to matrix–matrix multiplications where the MVTU is fed by a sliding window unit (SWU), a special stream buffer that enables windowed access to the input feature map while minimizing memory requirements.

The MVTU is parameterized in terms of input, output, and weight precision, as well as the type of resource [e.g., look-up tables (LUTs) or block memory (BRAM)] used for its internal weight and threshold memories. Furthermore, the MVTU can be parallelized in various dimensions limited only by the DNN topology and the available programmable logic resources. The dimensions comprise P parallel processing elements (PEs), which determine the number of output channels processed in parallel, and S single-instruction–multiple-data (SIMD) input channel lanes for each PE. To further scale small DNNs, such as the ones for RadioML, we extend FINN to support an additional degree of parallelism by processing M output positions simultaneously.

FINN Compiler Tool Flow

Figure 3 shows the FINN tool flow, which has a modular structure that allows the user to interactively generate a specialized architecture for a specific DNN. The framework provides a front-end, the FINN compiler with its transformation and analysis passes, and high-level synthesis (HLS)-based back-end to explore the design space in terms of resource and performance constraints. While users can build a custom step-by-step flow using the provided infrastructure, FINN also offers an automatic build flow that optimizes common DNN topologies based on a performance target and device constraints. Internally, FINN is built around an end-to-end intermediate representation (IR) based on the open neural network exchange (ONNX) format for DNN graphs. The IR also serves as the input format for quantized models, which are exported from a training front-end, such as Brevitas.

Starting from the input model, the FINN compiler performs three phases of graph transformation passes, which analyze and change the IR to gradually map it to a synthesizable accelerator architecture. In the preparation

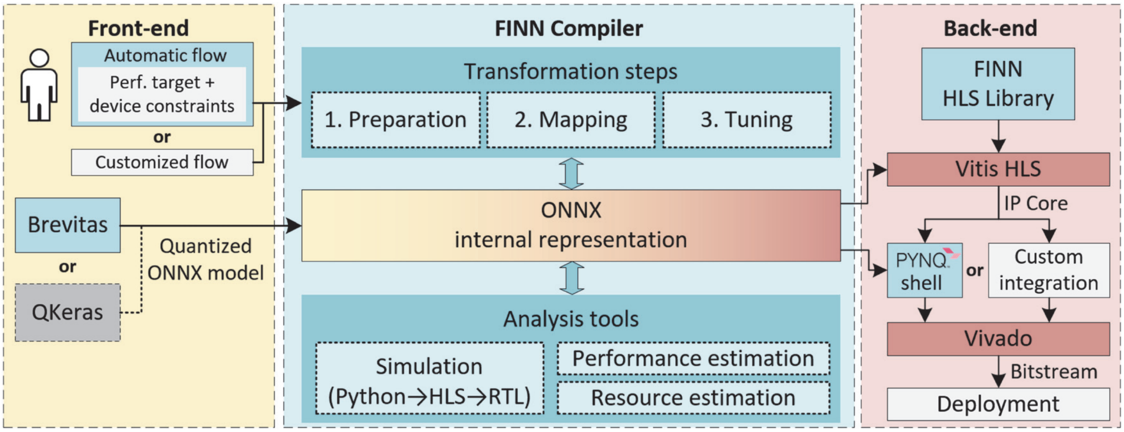


FIGURE 3. Overview of the FINN compiler flow.

phase, DNN graph nodes are rearranged or fused to make them compatible with how the back-end building blocks operate, for example, in terms of data layout. This includes convolution lowering and a "streamlining" process to merge quantization and batch-normalization factors into multithreshold operations. The mapping phase associates layers with the configurable operators implemented by the FINN HLS library, such that each node corresponds to a Vitis HLS C++ function call, which can later be synthesized to an IP block. In the tuning phase, the resource and parallelism configurations of the MVTUs are determined. A "folding" process assigns compute resources via a selection of P, S, and M to each layer to obtain the desired throughput within a balanced pipeline. Bottlenecks due to bursty behavior are avoided by automatically inserting stream buffers. FINN employs various analysis tools to guide the mapping phase. These include model-based performance and resource estimates, as well as simulation and reporting on all abstraction levels.

Finally, FINN generates code from the IR, synthesizes, and stitches the layers together. The resulting standalone IP core can be integrated into any design or deployed quickly with the generated shell project and driver for Xilinx Alveo and PYNQ platforms.

Brevitas

Brevitas[5] is a PyTorch extension for neural network quantization, with a focus on quantization-aware training (QAT). It provides building blocks to model a reduced precision inference data path at training time. Due to its flexibility, DNN models can be adopted to target different styles of fixed-point computing. By accounting for the additional error introduced by quantization at training time, QAT provides superior results in terms of accuracy compared to post-training quantization approaches and can gracefully scale the precision of both parameters and activations down to binary values.

For a given target datatype, Brevitas exposes multiple hyperparameters that a user can tune to adjust the quantization algorithm to the particular training problem at hand. For example, the scale factor of a given datatype, which for traditional fixed-point datatypes is a power-of-two number, can be set to a user-defined constant, a user-initialized value learned with backpropagation, or a value initialized according to some statistics and then learned with backpropagation.

Once the network has been trained, Brevitas can export it to a downstream toolchain by encoding it in an intermediate format. For use in the FINN framework, Brevitas extends ONNX by introducing *ad hoc* quantization nodes to specify custom fixed-point datatypes.

MODULATION CLASSIFICATION CASE STUDY

We choose the automatic modulation classification use case to showcase the potential of the FINN approach for RadioML.

Models and Training

We train our models on the RadioML 2018 dataset, which contains signals in 24 different modulation schemes at an SNR range from −20 dB to +30 dB. To limit the design space and provide a better comparison with related work, we stay very close to the "VGG10" topology proposed alongside the dataset by DeepSig.[3] This DNN consists of seven one-dimensional convolution layers with a kernel size of three, each followed by batch-normalization, ReLU activation, and max-pooling to reduce the output feature map size by half. This CNN block is followed by two

TABLE 1. Results of FINN prototypes against existing FPGA implementations. Utilization for XCZU28DR device.

Implementation	FINN A	FINN B	FINN C	Best from Tridgell et al.[6]	Best from den Boer et al.[7]
Topology (F_c, F_d)	VGG10 (64, 128)	VGG10-S (32, 128)	VGG10-S (32, 128)	VGG10-L (128, 512)	Other 1D-Conv
# Parameters	161,000	72,000	72,000	636,000	14,000
Quantization (weight/activation)	4 bits (5-bit first layer)	4 bits	4 bits	weights: ternary activations: mixed	6 bits
Frequency	250 MHz	250 MHz	250 MHz	250 MHz	250 MHz
LUTs (util.)	267,000 (63%)	65,000 (15%)	229,000 (54%)	211,000 (50%)	106,000 (25%)
Flip-flops (util.)	120,000 (14%)	42,000 (5%)	131,000 (15%)	324,000 (38%)	61,000 (7%)
BRAM blocks (util.)	56 (5%)	25 (2%)	26 (2%)	512 (47%)	0 (0%)
DSP slices (util.)	0 (0%)	0 (0%)	0 (0%)	1407 (33%)	137 (3%)
Accuracy @ 30 dB	94.1%	91.0%	91.0%	80.2%	71.8%
Throughput [samples/s]	246 million	246 million	1750 million	500 million	250 million
Latency [μs]	11.7	11.3	2.6	8.0	4.6

dense layers and a final dense classification layer. Besides the weight and activation quantization, we adjust only the number of filters in the convolution layers (F_c) as we found this to be the second-most effective method for trading off accuracy and compute cost.

For all models, we quantize inputs to a fixed 8-bit range determined by statistical analysis of the dataset to yield a low quantization error at high SNR ($\geq$ 6 dB) for most modulations. Two variants of single-sideband amplitude modulation deviate from the Gaussian-like distribution of other modulations and perform somewhat worse with quantization, which we deem an acceptable tradeoff. We observe that the relative SNR of the data that the network is trained and tested on has a large impact on recognition performance. While low-SNR environments are important for practical applications, we focus on training and testing on high-SNR data for brevity and to facilitate comparisons to related work.

During training, we approximate the compute cost for each model by calculating the number of "bit-operations" (BOPS) as a sum of all multiply-and-accumulate operations weighted by their respective operand bit-widths since this metric exhibits an approximately linear relationship to resource consumption of the resulting FINN accelerators under the same folding configuration. We find that quantizing weights and activations for the original VGG10 topology with $F_c = 64$ below a bit-width of 4 does not result in a compelling utilization-accuracy tradeoff. Instead, decreasing the number of convolution filters to $F_c = 32$ yields better accuracy and less cost than the original VGG10 quantized to 2-bit weights and activations, even if 4-bit precision is kept for the more quantization-sensitive input CNN layer. We refer to this smaller model variant as "VGG10-S" and select it alongside the original model for accelerator generation.

Prototype Results

Based on the two models VGG10 and VGG10-S, we build three distinct accelerator prototypes, each in a different corner of the vast design space. We target the XCZU28DR RFSoC device found on the ZCU111 and PYNQ RFSoC 2×2 development boards. Table 1 shows key metrics of the prototypes. Prototype "FINN A" is based on VGG10 and represents the most accurate—but resource-intensive—implementation with a peak accuracy of 94.1%, a 2.6 p.p. drop from the floating point (FP) baseline we trained to 96.7% accuracy. As for performance, FINN is configured to apply full parallelism across the input and output channel dimension, limiting throughput to one sample per FPGA clock cycle, with an actual throughput that is slightly lower (246 MS/s at 250 MHz) due to padding and pipeline inefficiencies. Note that we report throughput in samples per second instead of frames per second to decouple it from the frame size, which is 1,024 as in training.

"FINN B" is the smallest prototype and applies the same configuration to the VGG10-S model, resulting in the same performance, but lower accuracy at 91.0%. For "FINN C," we scale up the parallelism by extending

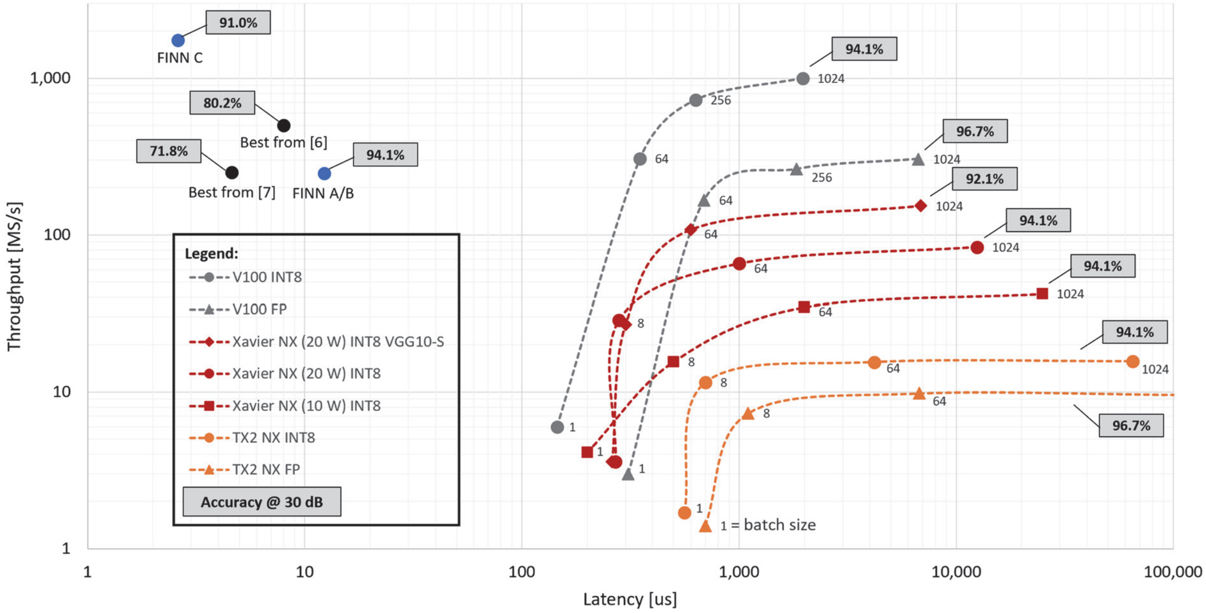


FIGURE 4. Throughput versus latency of VGG10 inference on various platforms. GPU results are batch-size-dependent and report end-to-end host latency. FPGA results assume a direct input data feed and report compute latency.

FINN to allow simultaneous processing of up to 8 samples/cycle. This unlocks the unprecedented performance of 1.75 GS/s at a latency of just 2.6 μs. This increases resource efficiency (throughput over utilization) by a factor of 2. We attribute this nonlinear scaling mainly to synthesis optimization: due to the pooling structure, a balanced pipeline that takes in multiple samples simultaneously has to apply full channel unfolding to more layers, which allows for the elimination of PE multiplexing logic and zero-weight multiplications. This results in a device utilization of 54%, with significant resources still available for additional features or performance scaling. In general, adjusting parallelism and bit-width alone can scale the implementation to performance, resource, or accuracy targets well beyond what we show here.

Table 1 also includes top-performing accelerators from related work that target the same device family. The implementation described in Tridgell et al.[6] uses a larger variant of the model with more convolution and dense filters (F_d) but applies harsh quantization with ternary weights and mixed activation formats. The DNN is mapped to hardware using a custom HDL-generation framework. The reported peak accuracy is significantly lower than our results and the design manages only 2 samples/cycle, despite the relatively high resource consumption.

In contrast, the prototype shown in den Boer et al.[7] implements a custom HLS-based mapping tool and focuses on smaller CNN topologies, but uses 6-bit operands. Even for their largest model, accuracy is low (71.8%) and the reported throughput is 1 sample/cycle. Further related work is discussed in the sidebar.

GPU Comparison

To compare performance and energy efficiency with current GPUs, we use NVIDIA's TensorRT tool to run automatically optimized inference benchmarks on the Tesla V100 data-center GPU and two modules of the Jetson embedded GPU family: the previous-generation TX2 and the current-generation Xavier NX, which we run in its lowest (10 W) and highest (20 W) power modes. The Xavier NX also features dedicated INT8 compute support for more efficient DNN inference. We utilize this by applying 8-bit post-training quantization to the FP models using TensorRT, incurring an accuracy drop from 96.7% to 94.1% for VGG10 and from 95% to 92.1% for VGG10-S, although this may be alleviated via QAT.

Figure 4 shows the resulting throughput over latency, both in logarithmic scale, against the discussed FPGA accelerators. For the GPUs, we report end-to-end latency, which includes data transfer and synchronization overhead that typically ranges from 1% to 10% in this case. While the streaming accelerators take in samples as they are supplied from the digital radio front-end, and do not even need to buffer a single frame before computation begins, GPUs require

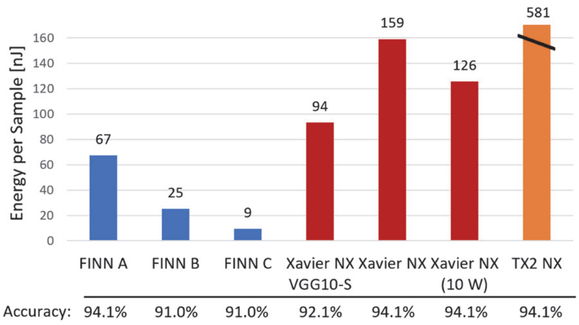


FIGURE 5. Power efficiency of FINN accelerators versus NVIDIA Jetson GPU platforms (batch-64, INT8).

the aggregation of frames into batches to utilize their computing power. This is evident in our measurements, as peak throughput (153 MS/s for Xavier NX) and minimum latency (260 μs for Xavier NX) are not achievable at the same time. Regardless, even this latency is 100× higher than that of our prototype FINN C. When comparing our solution to devices of a similar power envelope, i.e., the embedded GPUs, it becomes clear how only the FPGA streaming accelerator is capable of keeping up with the RF data rate of multiple gigasamples per second, all while delivering exceptional microsecond latency. We expect the direct RFSoC integration to only amplify this advantage for real-world systems, where additional overhead will be needed to feed signals to GPUs.

Figure 5 compares measured power efficiency in terms of energy per processed sample of FINN accelerators and embedded GPUs. Our fastest and most efficient prototype consumes 16.5 W and is 17× more power efficient than the INT8 VGG10 on Xavier NX with 10.5 W, albeit with 3.1 p.p. lower accuracy. For a fairer comparison, we also run VGG10-S on the Xavier NX. Even if we assume that quantization-aware training could improve the INT8 accuracy to FP level, FINN A would lie within one percentage point of accuracy while delivering 1.4× the efficiency and 2.3× the throughput.

Scalability

To demonstrate FINN's scalability beyond the accuracy-optimized prototypes discussed before, we synthesize accelerators for larger VGG10 instances and two additional topology families: VGG24, a deeper variant of VGG10 with 3× the convolution layers, and "BacalhauNet,"[8] the winning DNN of the "Lightning-Fast Modulation Classification" problem statement of the 2021 ITU AI in 5G Challenge, which features depthwise-separable convolutions with wide kernel dimensions and residual connections. We measure DNN size as inference compute cost per classification (in BOPS) and scale it for each topology family via the number of convolution filters. As a third variable, the generated accelerators target two levels of throughput. Figure 6 plots the resulting FPGA utilization, showcasing the proportional relationship between accelerator LUT count and DNN size.

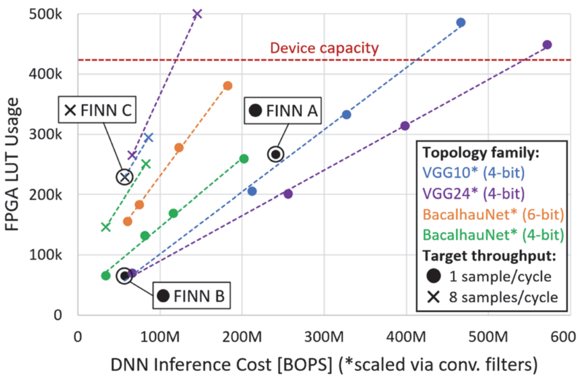


FIGURE 6. Scaling to larger DNNs and different topologies. Each data point represents a FINN-generated accelerator for a DNN of the specified topology family and inference cost, which is scaled along the *x*-axis by increasing the number of convolution filters. Our three accuracy-optimized prototypes (A, B, and C implementations from Table 1) are marked as singular experiments for reference.

> *AVOIDING OBSTRUCTIVE MANUAL DESIGN EFFORTS, THE OPEN-SOURCE FINN FRAMEWORK AUTOMATICALLY GENERATES DNN ACCELERATORS AND REACHES UNPRECEDENTED RESULTS IN ALL RELEVANT METRICS, INCLUDING ENERGY EFFICIENCY.*

In summary, VGG24 scales best to larger models, especially when compared to the original 6-bit BacalhauNet. A more sensible 4-bit variant performs close to VGG10 at a nominal throughput of 1 sample/cycle and scales just as well to the extremely parallel 8 samples/cycle configuration, reaching the real-world performance of 1.8-GS/s and 1.5-μs latency at 250 MHz. In some scenarios, the DNN scaling method does not quite allow for full device utilization due to underlying tool limitations. A fair power comparison against INT8 TensorRT execution on the Xavier NX platform is

RELATED WORK

SDR systems with hardware acceleration are an emerging field. Deepwave Digital released the first commercial "AI Radio Transceiver" (AIR-T)[9] in early 2020, a device that integrates a transceiver, an FPGA, and an NVIDIA Jetson TX2 embedded GPU with a GNU-Radio software stack. The FPGA performs necessary DSP tasks and acts as a bridge between the serial transceiver interface and the GPU, which is responsible for DNN inference. The system operates in the 100–200-MS/s range and relies on the shared-memory architecture of modern embedded GPUs to minimize latency, as no additional memory copy between the host (CPU) and GPU is necessary. In contrast, we combine all functions into a single SoC and remove even the last remaining external memory transfer between the transceiver and the accelerator. We also address an often mentioned argument against FPGA-based solutions, the high development effort, with our automated FINN tool flow.

Regarding the use case modulation classification, we highlight two state-of-the-art FPGA accelerator approaches[6,7] in Table 1. Other FPGA implementations for modulation classification exist and include the use of support vector machines[10] for classification, DNN processing of engineered statistical features,[11] and even spiking neural networks.[12] However, most of these approaches are not comparable to our work, e.g., due to undisclosed data sets, or lack of a flexible toolchain and meaningful GPU comparison. On the training side, efforts have been made to create more resource-efficient DNN topologies through the use of sparsity, residual connections, depthwise convolutions, or recurrent neural networks. We refer to Jdid et al.[13] for a survey.

difficult without in-depth accuracy testing, but results from our scaling experiments continue to suggest a strong efficiency advantage for FINN, which ranges from 3.4× for the largest VGG24 over 4.2× for the largest 6-bit BacalhauNet to around 30× for the fast 4-bit variants. In terms of raw performance, even the smallest BacalhauNet does not surpass 65-MS/s throughputs and 1-ms latency (batch-64) on the Xavier NX.

CONCLUSION

On the basis of modulation classification, we have shown how FPGA streaming architectures suit DNN-based RF signal processing perfectly, especially in RFSoC systems with integrated radio front-ends. Avoiding obstructive manual design efforts, the open-source FINN framework automatically generates DNN accelerators and reaches unprecedented results in all relevant metrics, including energy efficiency. Compared to current embedded GPUs, we achieve orders of magnitude better latency (microseconds versus milliseconds) and throughput (gigasamples per second versus hundreds of megasamples per second), while keeping the quantization-induced accuracy penalty under control. Future work includes the implementation of a live RFSoC demonstrator and exploration of promising DNN topologies and techniques, such as ResNets and sparsity.

REFERENCES

1. T. J. O'Shea, K. Karra, and T. C. Clancy, "Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol.*, 2016, pp. 223–228.
2. T. Erpek, T. J. O'Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, "Deep learning for wireless communications," 2020, *arXiv:2005.06068*.
3. T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.
4. M. Blott et al., "FINN-R: An end-to-end deep-learning framework for fast exploration of quantized neural networks," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 11, no. 3, Dec. 2018, Art. no. 16.
5. A. Pappalardo, "Brevitas (software)." Accessed: Aug. 22, 2022. [Online]. Available: https://doi.org/10.5281/zenodo.5779154
6. S. Tridgell, D. Boland, P. H. Leong, R. Kastner, A. Khodamoradi, and Siddhartha, "Real-time automatic modulation classification using RFSoC," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops*, 2020, pp. 82–89.
7. H. den Boer, R. Muller, S. Wong, and V. Voogt, "FPGA-based deep learning accelerator for RF applications," in *Proc. IEEE Mil. Commun. Conf.*, 2021, pp. 751–756.

8. J. Rosa et al., “BacalhauNet: A tiny CNN for lightning-fast modulation classification,” *ITU J. Future Evolving Technol.*, vol. 3, no. 2, pp. 252–260, 2022.
9. J. Ferguson, P. Witkowski, W. Kirschner, and D. Bryant, “Deepwave digital creates an AI enabled GPU receiver for a critical 5G sensor,” white paper, 2020. [Online]. Available: https://developer.nvidia.com/blog/wp-content/uploads/2020/01/NVIDIA_Blog_v2.pdf
10. C. Cardoso, A. R. Castro, and A. Klautau, “An efficient FPGA IP core for automatic modulation classification,” *IEEE Embedded Syst. Lett.*, vol. 5, no. 3, pp. 42–45, Sep. 2013.
11. A. F. de Castro, R. S. R. Milléo, L. H. A. Lolis, and A. A. Mariano, “Artificial neural network based automatic modulation classification system applied to FPGA,” in *Proc. ACM Symp. Integr. Circuits Syst. Des.*, 2021, pp. 1–6.
12. A. Khodamoradi, K. Denolf, and R. Kastner, “S2N2: A FPGA accelerator for streaming spiking neural networks,” in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, 2021, pp. 194–205.
13. B. Jdid, K. Hassan, I. Dayoub, W. H. Lim, and M. Mokayef, “Machine learning based automatic modulation recognition for wireless communications: A comprehensive survey,” *IEEE Access*, vol. 9, pp. 57851–57873, 2021.

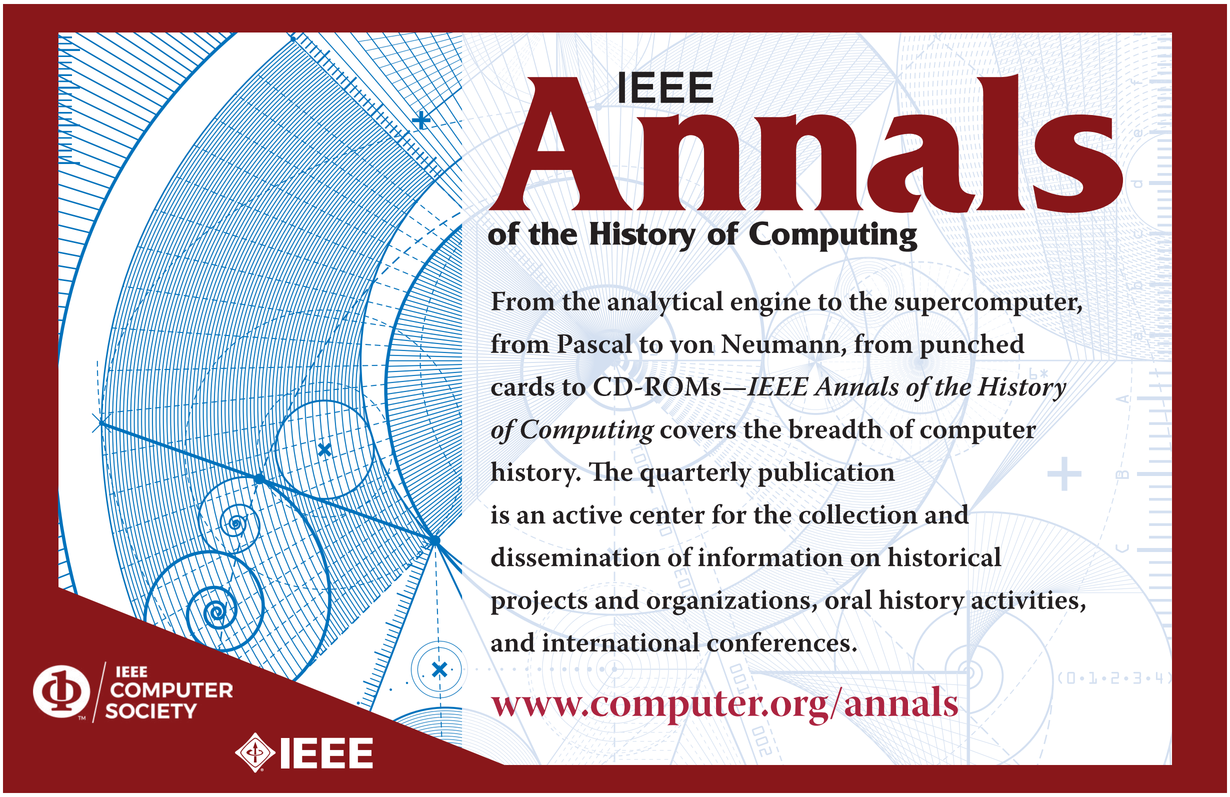
FELIX JENTZSCH is a Ph.D. student with Paderborn University. His research focuses on automated hardware/software co-design for reconfigurable computing systems. Jentzsch received a master’s degree in computer engineering from Paderborn University. He is a Member of IEEE. Contact him at felix.jentzsch@upb.de.

YAMAN UMUROGLU is a senior MTS with AMD, Dublin, Ireland. His research interests include full-stack view of DNNs with a focus on high efficiency and spans hardware-network co-design, techniques for efficient arithmetic, sparsity and quantization. Contact him at yamanu@amd.com.

ALESSANDRO PAPPALARDO is a staff researcher with AMD, Dublin, Ireland. His research interests include neural network co-design and acceleration on reconfigurable hardware. Contact him at alessand@amd.com.

MICHAELA BLOTT works as a Senior Fellow with AMD, Dublin, Ireland. Her research interests include compute architectures, reconfigurable computing, and machine learning. She is a Member of IEEE. Contact her at mblott@amd.com.

MARCO PLATZNER is a professor for computer engineering with Paderborn University, Paderborn, Germany. His research interests include reconfigurable computing, hardware-software co-design, and parallel architectures. He is a Senior Member of IEEE. Contact him at platzner@upb.de.



EDITORS: Barry I. Schneider, bis@nist.gov
Gabriel A. Wainer, gwainer@sce.carleton.ca

DEPARTMENT: COMPUTER SIMULATIONS

This article originally appeared in
computing in SCIENCE & ENGINEERING
vol. 24, no. 5, 2022

Hybrid Models That Combine Machine Learning and Simulations

Philippe J. Giabbanelli, *Miami University, Oxford, OH, 45056, USA*

Simulation experts are now well acquainted with machine learning (ML) techniques, using them to find patterns in data that can later be turned into rules of a simulation or enabling their simulated entities to adapt and learn. In the other direction, ML experts occasionally make use of simulated data to create controlled experiments in which learning algorithms can be evaluated. In this article, we go beyond these typical uses by focusing on current opportunities that have the potential to bring the two research communities together. These opportunities can be realized in areas where the potential of hybrid ML/simulation methods has not been fully attained yet. Such applications also motivate the development of innovative methods, for example, to combine the accuracy of ML with the interpretability of simulation models. Using select examples from our interdisciplinary team, this article reflects on opportunities in applications and techniques to promote productive conversations across research areas.

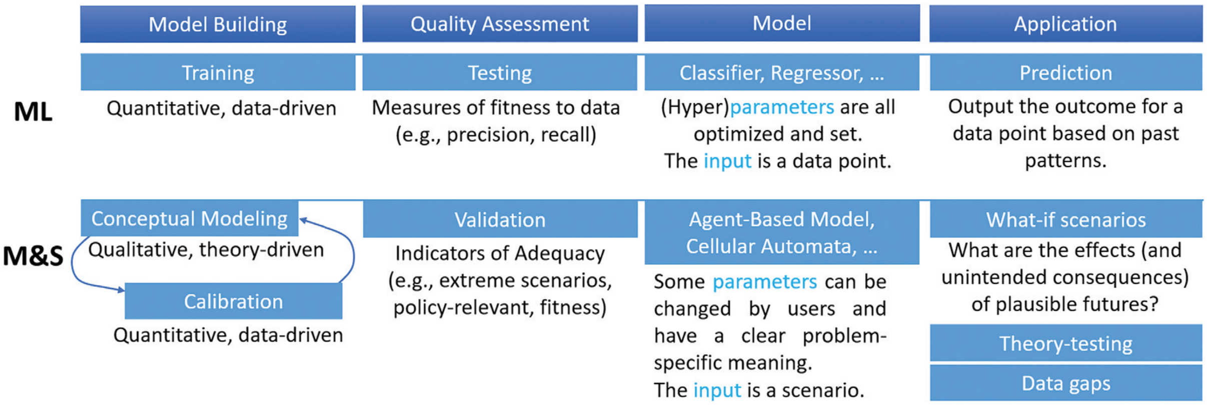
The explosion of machine learning (ML) research has permeated virtually all research areas, and modeling and simulation (M&S) is no exception. At a high level, the two fields apply similar processes: both seek to create *models* that can make predictions on new *instances*, and both involve a part of the data for *model building* and a separate part for *quality assessment* (see Figure 1). However, details reveal important differences for each of these parts.

M&S involves a *mix* of theories and data to analyze the behavior of a system over time, either for explanatory or predictive purposes. In contrast, ML leverages past data to create predictive statistical models (e.g., classifier and regressor), generally on the assumption that they follow the same patterns as observed in the past. For example, we can ask a simulation model to examine a new scenario (known as a "what-if analysis") that depicts a very different future, such as significant public health interventions. This situation would be challenging for an ML model since it was tuned automatically from past data, and users cannot easily change it to reflect potential futures.

This divergence between future-oriented scenarios and replicating past patterns is also reflected in the model-building process: simulations often involve interdisciplinary teams to craft rules based on *theories* and refine them from data, whereas ML is data-centric. Theories are particularly important for simulation models in the social sciences, where ML can be seen as "theoryless big data modeling" and struggles to answer intervention-focused questions.[1] Simulation models promote transparency for co-design with domain experts and/or flexibility to examine new scenarios, whereas ML models aim at maximizing a fitness measure (e.g., accuracy, precision, or recall). The quality of a simulation model may, thus, be lower than that of ML with respect to fitting the data, but it may still be *adequate* or "fit for purpose" if it satisfies goals other than predictions, such as identifying unintended consequences or guiding data collection efforts.

Despite these fundamental differences, *hybrid* approaches have emerged at the intersection of ML and M&S. Research terminology[2] distinguishes hybrid simulation (when several simulation techniques are used together) from *hybrid modeling* (when simulation techniques are combined with approaches from other fields, such as ML). In this article, we focus on opportunities for hybrid modeling. Growth in this area has partly served to address two needs.

1521-9615 © 2023 IEEE
Digital Object Identifier 10.1109/MCSE.2023.3238649
Date of current version 1 May 2023.



	Model Building	Quality Assessment	Model	Application
ML	Training Quantitative, data-driven	Testing Measures of fitness to data (e.g., precision, recall)	Classifier, Regressor, ... (Hyper)parameters are all optimized and set. The input is a data point.	Prediction Output the outcome for a data point based on past patterns.
M&S	Conceptual Modeling Qualitative, theory-driven Calibration Quantitative, data-driven	Validation Indicators of Adequacy (e.g., extreme scenarios, policy-relevant, fitness)	Agent-Based Model, Cellular Automata, ... Some parameters can be changed by users and have a clear problem-specific meaning. The input is a scenario.	What-if scenarios What are the effects (and unintended consequences) of plausible futures? Theory-testing Data gaps

FIGURE 1. Although there are high-level procedural similarities between machine learning (ML) and modeling and simulation (M&S), each step has noticeable differences. Although the process is presented linearly, note that revisions are frequent; hence, every step can go back (e.g., if a model does not satisfy the needs of quality assessment, then we go back to model building).

First, as new ML algorithms are developed, there is a need to evaluate them across various settings (e.g., given certain levels of imbalance in the data, noise, or outliers). Simulations, via their meaningful parameters, can create synthetic data to use in evaluating ML algorithms in a controlled setting. This can support critical applications with scarce data, such as detecting malfunctions at a nuclear facility, which has not happened frequently; hence, detection systems lack training data. In the other direction, there is a realization that not *every* part of a simulation model needs to be transparent and flexible. For example, an agent-based simulation model for the spread of COVID-19 may include several policy-relevant parameters (e.g., mask mandates and vaccine capacity) and use select theories to represent compliance. Simulated individuals are also observing and adapting (i.e., *learning*), and this necessary component of the model can be achieved by ML. In this context, the training data may originate from classic sources for ML, such as surveys or surveillance data, resulting in a model that is built at the *beginning* of the simulation and potentially updated as the simulation is executed.

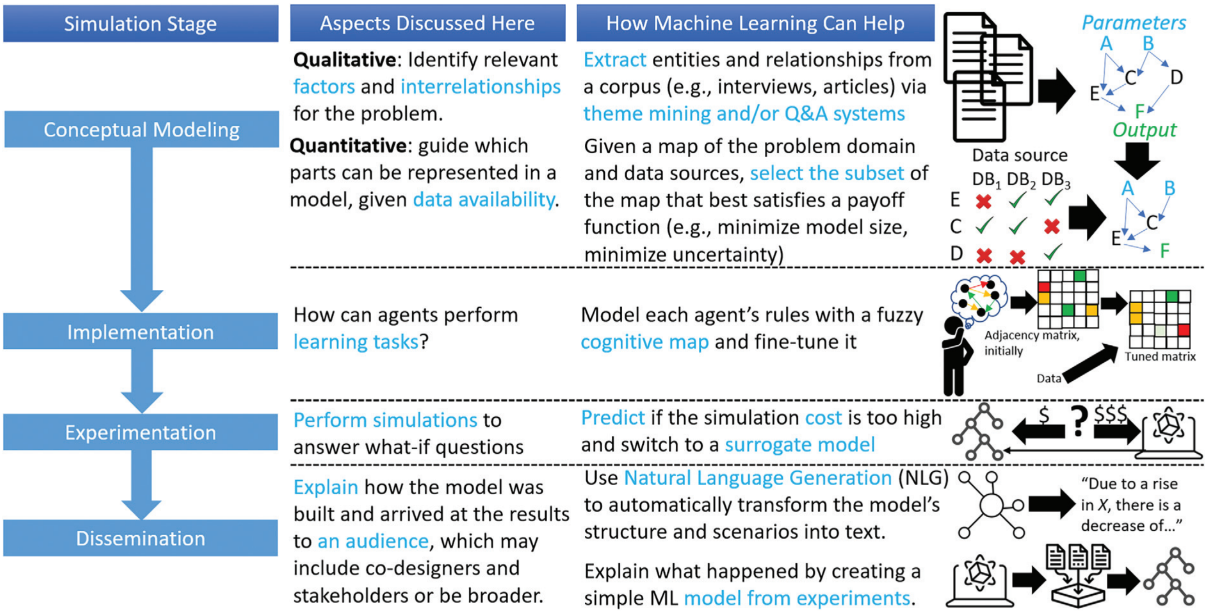
Alternatively, the training data may come solely from the simulation,[3] with each agent continuously observing its peers and forming rules about their behavior (e.g., to associate certain observable traits in peers with an outcome). This notion of equipping simulated agents with learning algorithms is one of the most common manifestations of hybrid models and is occasionally referred to as "heuristic rule-based models."[4]

The focus of ML algorithms on accuracy can, thus, serve to replace or complement specific stages of the simulation process, which is the focus of this article. Each of the following sections is dedicated to the potential use of ML for a specific stage of the simulation process, following the order shown in Figure 2.

CONCEPTUAL MODELING: WHAT SHOULD GO INTO A MODEL?

A conceptual model is used at the early stages of a modeling project to offer a simple representation of a system, often in a graphical format, by listing salient concepts and interrelationships. For example, a conceptual model for the spread of COVID-19 may state that beliefs and attitudes about vaccination depend on political leanings, but it may not go as far as to include concepts and interrelationships showing that people favor certain political stances. This illustrates that a conceptual model is necessarily a *simplification* of reality, as some aspects are deemed out of scope by modelers and/or stakeholders, not sufficiently understood to be modeled, or lack data.

Despite the importance of a conceptual model in setting the agenda for the rest of the process, factors and interrelationships are often summarily disclosed by modelers as the "result" of an unknown discussion within their interdisciplinary team. For example, there is *limited disclosure* about how different perspectives are elicited and integrated.[5] In a previous study, we found that modelers are not neutral: when working with the *same* evidence base, they can produce *different* conceptual models[6] and, hence, arrive at different simulation results. Even if every modeler shared their code (which is far from being the case), issues of *provenance* would not be resolved since we would only see the resulting conceptual model without knowing the process that created it.



Simulation Stage	Aspects Discussed Here	How Machine Learning Can Help
Conceptual Modeling	**Qualitative**: Identify relevant factors and interrelationships for the problem.	Extract entities and relationships from a corpus (e.g., interviews, articles) via theme mining and/or Q&A systems
	Quantitative: guide which parts can be represented in a model, given data availability.	Given a map of the problem domain and data sources, select the subset of the map that best satisfies a payoff function (e.g., minimize model size, minimize uncertainty)
Implementation	How can agents perform learning tasks?	Model each agent's rules with a fuzzy cognitive map and fine-tune it
Experimentation	Perform simulations to answer what-if questions	Predict if the simulation cost is too high and switch to a surrogate model
Dissemination	Explain how the model was built and arrived at the results to an audience, which may include co-designers and stakeholders or be broader.	Use Natural Language Generation (NLG) to automatically transform the model's structure and scenarios into text. Explain what happened by creating a simple ML model from experiments.

FIGURE 2. We identify opportunities to use or develop ML techniques in each of the four stages of the M&S process. Each opportunity is discussed in a dedicated section, starting with conceptual modeling and its two core tasks. Although there exist more stages (e.g., calibration, verification, and validation) and tasks, this short article necessarily focuses on a subset. Q&A: question-and-answer.

However, it is important to ensure that recent evidence is adequately taken into account and that measures are in place to limit biases. For example, issues can arise when potential users of a model to manage the impact of the spread of COVID-19 on hospitals are unaware that it utilizes evidence from two years ago (which may be deprecated when dealing with a rapidly changing situation) or only relies on interviews with doctors and nurses, hence potentially missing complementary perspectives related to health equity and psychological well-being. To ensure that other modelers and potential end users are clearly informed about the evidence base that goes into a model and how it is utilized, this section details the possibility of switching from current person-dependent and/or undisclosed methods to ML.

At the qualitative stage of conceptual modeling, the goal is to identify potential concepts and interrelationships. In a nontransparent process, they may be identified by a modeler's own assumptions or a brainstorming effort by a specific team (which would not be reproducible by another team). ML can make this process more transparent by specifying how factors and interrelationships are extracted (e.g., via theme mining or Q&A systems) from a text corpus consisting of stakeholder and participant interviews and/or authoritative documents. Although ML could entirely automate the process of building a conceptual model, the objectives of the model and prevailing theories may already suggest some salient factors and interrelationships to the modelers. For example, when creating a model for suicide prevention, it is a given that suicidal thoughts must come before attempts and then death. Hence, a part of the model may be set, while the rest can be discovered via ML. In addition, given a conceptual model, ML can automatically contrast it with social media posts as a means to elicit the views of a population that may eventually be impacted by decisions from the model.

A conceptual model is akin to a road map: it shows potential paths and important locations, but it does not tell us where to go. That decision ultimately lies with the modeling team, based on considerations such as data availability. Indeed, if we make a model based on mechanisms for which we have no data, then there is high uncertainty about the outcomes. For example, we know that asymptomatic cases, vaccination, and social distancing have an effect on disease prevalence. However, we may not have data about these mechanisms, such as the number of asymptomatic cases, the efficacy of vaccination, or the average level of social distancing. Given these unknowns, the simulated disease prevalence may be subject to a wide confidence margin. In a previous study, we proved that minimizing uncertainty by selecting the best data sources is an

NP-complete problem[7]; hence, modeling teams must resort to heuristics, such as keeping their models simple. The transformation of a conceptual model into an operational one based on data is often shrouded in mystery, as teams do not always state which heuristic is used and as they lack algorithmic means of automatically navigating data sources for a given heuristic. Again, ML has the potential to help address these gaps.

IMPLEMENTATION: HOW WILL AGENTS LEARN?

In an agent-based model specifically, agents can adapt by observing their peers or the environment and adjusting their actions accordingly. Ten years ago, common approaches would fine-tune handcrafted rules (e.g., thresholds in if–then rules or components of the belief–desire–intention framework); employ cognitive architectures; or venture into simple ML models, such as decision trees.[8] With a rise in ML, we can now go beyond fine-tuning parameters since ML approaches create the rules themselves. As we noted previously,[9] ML can thus tackle the matter of structural uncertainty (which rules should govern an agent's behavior) instead of being limited to parametric uncertainty (adjusting values in an established rule set).

There are now several approaches that allow agents to make observations and derive a model from them. Two diametrically opposed approaches to embed a "virtual brain" in each agent[10] are exemplified by deep neural networks (DNNs) (which can provide accuracy at the expense of transparency and computational cost) and fuzzy cognitive maps (FCMs) (which seek transparency and are computationally light but potentially less accurate). FCMs are an interesting case at the crossroads of cognitive architectures and ML. They are traditionally elicited from individuals as a means to externalize their "mental model" into a directed, weighted, labeled network. However, emerging ML algorithms are gradually making it possible to dynamically change relationships in the agents' mental models. As advances have made it possible to equip agents with ML models, the research frontier is shifting from the individuals (e.g., enabling each agent to learn via a DNN) to the collective (e.g., how do agents behave as a group through their respective ML models?).

EXPERIMENTATION: WHEN SIMULATIONS BECOME WASTEFUL

Several COVID-19 simulation models have been able to use computational resources specifically set aside by consortia. These models, thus, often require high-performance computing clusters to run simulations. However, the *costs* of simulations are not identical: some of them can terminate sooner, for example, when it becomes sufficiently clear that a disease will spread. In addition, results can be *related*: if a population with low vaccine coverage gets sick, then it will still be the case with lower coverage. Based on these two observations, we suggested a two-step process for ML and M&S[9]: an ML model can predict the cost of performing a simulation and, if it exceeds a user-defined limit or resources available at the time, results would be obtained by a *surrogate* (i.e., an ML model trained from past simulation results) instead of by performing a new simulation. When simulations can be afforded, they will grow the data from which the surrogate is trained, thus improving its fidelity. Although neither of these two steps is technically new, they were seldom used in COVID-19 models, hence leading to computational costs that could have been reduced drastically.[11] There is, thus, room for improvement in the uptake of practices from one community to the other.

DISSEMINATION: TOWARD EXPLAINABLE SIMULATIONS

"You should do it because the simulation results say so" is no longer a sufficiently convincing argument—if it ever was one. Explaining how a model was built and distilling its results is, thus, a growing necessity, both in ML (as part of explainable artificial intelligence) and in simulations. Although we may expect simulations to be at an advantage for explanations since they are (partly) built by humans and involve theories, experiences from COVID-19 have shown that explaining can be a difficult task for modelers, particularly when the target audience represents a broad range of skills and backgrounds. Perhaps counterintuitively, ML can be the key to automatically explaining simulations at two levels.

First, we need to explain the *conceptual model*, which can involve many rules and parameters. Diagrammatic representations may not be easily understood outside of a community of practice. Reports (e.g., written under the protocol "overview, design concepts, and details") can be read by simulationists and domain experts alike, but they are manually written and, hence, can be subject to vagueness or miss certain aspects in addition to being laborious to write for complex models. Through advances in natural language generation via tools such as GPT-3, it is now possible to take a model (e.g., a large concept map) as input and eventually produce text. Experiments conducted by our research group have shown that this approach can generate *some* well-formed and varied sentences to convey a conceptual model,[12] although efforts are still needed to address errors in the output (e.g., misinterpretation).

Second, we need to synthesize the *results*. Having several scenarios and a large number of combinations of parameter values (e.g., for sensitivity analysis) can produce many results. While visualizations also have a role to play, ML can help us discern patterns when results from simulations form large datasets, thus contributing to streamlining the takeaway messages. Concretely, training an interpretable ML model (e.g., a decision tree) can reveal how a key simulation outcome is obtained. While deriving an ML model from a simulation is a common task (i.e., surrogate modeling), the emphasis is normally on providing a computationally cheaper proxy rather than supporting the interpretation of results. The interesting frontier in this regard is to promote a feedback loop whereby results are conveyed by an interpretable ML model, end users can highlight where they see discrepancies with their expectations, and we then identify where adjustments should be made in the simulation.

CONCLUSION

Based on our experiences, we have presented opportunities to either apply or design new techniques at the interface of M&S and ML. While we focused on four stages of the M&S process, the potential for ML is also present at several other stages, such as repurposing anomaly detectors to perform verification and validation. Although M&S practitioners occasionally question their role given the growth of ML, the opportunities presented here make the case for the numerous benefits that simulations can derive from improvements in ML. This article, thus, hopes to guide productive conversations and support synergistic efforts between research communities.

REFERENCES

1. W. Rand, "Theory-interpretable, data-driven agent-based modeling," *Social-Behavioral Model. Complex Syst.*, early access, Apr. 2019, doi: 10.1002/9781119485001.ch15.
2. N. Mustafee, A. Harper, and B. S. Onggo, "Hybrid modelling and simulation (M&S): Driving innovation in the theory and practice of M&S," in *Proc. IEEE Winter Simul. Conf. (WSC)*, Dec. 2020, pp. 3140–3151, doi: 10.1109/WSC48552.2020.9383892.
3. H. Kavak, J. J. Padilla, C. J. Lynch, and S. Y. Diallo, "Big data, agents, and machine learning: Towards a data-driven agent-based modeling approach," in *Proc. Annu. Simul. Symp.*, Apr. 2018, pp. 1–12.
4. L. An, "Modeling human decisions in coupled human and natural systems: Review of agent-based models," *Ecological Model.*, vol. 229, pp. 25–36, Mar. 2012, doi: 10.1016/j.ecolmodel.2011.07.010.
5. B. Hedelin *et al.*, "What's left before participatory modeling can fully support real-world environmental planning processes: A case study review," *Environ. Model. Softw.*, vol. 143, Sep. 2021, Art. no. 105073, doi: 10.1016/j.envsoft.2021.105073.
6. A. J. Freund and P. J. Giabbanelli, "Are we modeling the evidence or our own biases? A comparison of conceptual models created from reports," in *Proc. IEEE Annu. Model. Simul. Conf. (ANNSIM)*, 2021, pp. 1–12, doi: 10.23919/ANNSIM52504.2021.9552054.
7. A. J. Freund and P. J. Giabbanelli, "The necessity and difficulty of navigating uncertainty to develop an individual-level computational model," in *Proc. Int. Conf. Comput. Sci.*, Cham, Switzerland: Springer-Verlag, 2021, pp. 407–421, doi: 10.1007/978-3-030-77980-1_31.
8. W. G. Kennedy, "Modelling human behaviour in agent-based models," in *Agent-Based Models of Geographical Systems*, A. Heppenstall, A. Crooks, L. See, and M. Batty, Eds. Dordrecht, The Netherlands: Springer-Verlag, 2012, pp. 167–179.
9. P. J. Giabbanelli, "Solving challenges at the interface of simulation and big data using machine learning," in *Proc. IEEE Winter Simul. Conf. (WSC)*, 2019, pp. 572–583, doi: 10.1109/WSC40007.2019.9004755.
10. A. Negahban and P. J. Giabbanelli, "Hybrid agent-based simulation of adoption behavior and social interactions: Alternatives, opportunities, and pitfalls," *IEEE Trans. Comput. Social Syst.*, vol. 9, no. 3, pp. 770–780, Jun. 2022, doi: 10.1109/TCSS.2021.3101794.
11. C. B. Lutz and P. J. Giabbanelli, "When do we need massive computations to perform detailed COVID-19 simulations?" *Adv. Theory Simul.*, vol. 5, no. 2, Feb. 2022, Art. no. 2100343, doi: 10.1002/adts.202100343.
12. A. Shrestha, K. Mielke, T. A. Nguyen, and P. J. Giabbanelli, "Automatically explaining a model: Using deep neural networks to generate text from causal maps," in *Proc. Winter Simul. Conf.*, 2022, pp. 2629–2640.

PHILIPPE J. GIABBANELLI is with the Department of Computer Science and Software Engineering, Miami University, Oxford, OH, 45056, USA. Contact him at giabbapj@miamioh.edu.

Conference Calendar

IEEE Computer Society conferences are valuable forums for learning on broad and dynamically shifting topics from within the computing profession. With over 200 conferences featuring leading experts and thought leaders, we have an event that is right for you. Questions? Contact conferences@computer.org.

APRIL

9 April

- SaTML (IEEE Conf. on Secure and Trustworthy Machine Learning), Toronto, Canada

17 April

- COOL CHIPS (IEEE Symposium in Low-Power and High-Speed Chips), Tokyo, Japan

22 April

- VTS (IEEE VLSI Test Symposium), Tempe, Arizona, USA

23 April

- PacificVIS (IEEE Pacific Visualization Symposium), Tokyo, Japan

29 April

- DCOSS-IoT (Int'l Conf. on Distributed Computing in Smart Systems and the Internet of Things), Abu Dhabi, United Arab Emirates

MAY

1 May

- MOST (IEEE Int'l Conf. on Mobility, Operations, Services and Technologies), Dallas, USA

5 May

- ISPASS (IEEE Int'l Symposium on Performance Analysis of Systems and Software), Indianapolis, USA

6 May

- FCCM (IEEE Int'l Symposium on Field-Programmable Custom Computing Machines), Orlando, USA
- HOST (IEEE Int'l Symposium on Hardware Oriented Security and Trust), Tysons Corner, Virginia, USA
- ICFEC (IEEE Int'l Conf. on Fog and Edge Computing), Philadelphia, USA

13 May

- CCGrid (IEEE Int'l Symposium on Cluster, Cloud and Internet Computing), Philadelphia, USA
- ICCPS (ACM/IEEE Int'l Conf. on Cyber-Physical Systems), Hong Kong
- ICDE (IEEE Int'l Conf. on Data Eng.), Utrecht, The Netherlands
- RTAS (IEEE Real-Time and Embedded Technology and Applications Symposium), Hong Kong

20 May

- SP (IEEE Symposium on Security and Privacy), San Francisco, USA

21 May

- ISORC (IEEE Int'l Symposium on Real-Time Distributed Computing), Tunis, Tunisia

27 May

- FG (IEEE Int'l Conf. on Automatic Face and Gesture Recognition), Istanbul, Turkey
- ICST (IEEE Conf. on Software Testing, Verification and Validation), Toronto, Canada
- IPDPS (IEEE Int'l Parallel and Distributed Processing Symposium), San Francisco, USA

28 May

- ISMVL (IEEE Int'l Symposium on Multiple-Valued Logic), Brno, Czech Republic

JUNE

3 June

- ICHI (IEEE Int'l Conf. on Healthcare Informatics), Orlando, USA

4 June

- ICSA (IEEE Int'l Conf. on Software Architecture), Hyderabad, India
- WoWMoM (IEEE Int'l Symposium on a World of Wireless, Mobile and Multimedia Networks), Perth, Australia

10 June

- ARITH (IEEE Symposium on Computer Arithmetic), Malaga, Spain

16 June

- CVPR (IEEE/CVF Conf. on Computer Vision and Pattern Recognition), Seattle, USA

17 June

- SVCC (Silicon Valley Cybersecurity Conf.), Seoul, South Korea

2469-7087/24 © 2024 IEEE

19 June

- CHASE (IEEE/ACM Conf. on Connected Health: Applications, Systems and Eng. Technologies), Wilmington, USA

24 June

- DSN (IEEE/IFIP Int'l Conf. on Dependable Systems and Networks), Brisbane, Australia
- MDM (IEEE Int'l Conf. on Mobile Data Management), Brussels, Belgium
- RE (IEEE Int'l Requirements Eng. Conf.), Reykjavik, Iceland

25 June

- CAI (IEEE Conf. on Artificial Intelligence), Singapore

26 June

- CBMS (IEEE Int'l Symposium on Computer-Based Medical Systems), Guadalajara, Mexico

27 June

- CS (IEEE Cloud Summit), Washington, DC, USA (Hybrid)

29 June

- ISCA (ACM/IEEE Annual Int'l Symposium on Computer Architecture), Buenos Aires, Argentina

JULY

1 July

- COMPSAC (IEEE Annual Computers, Software, and Applications Conf.), Osaka, Japan
- ICALT (IEEE Int'l Conf. on Advanced Learning Technologies), Nicosia, Cyprus

3 July

- IOLTS (IEEE Int'l Symposium on On-Line Testing and Robust System Design), Rennes, France

7 July

- SERVICES (IEEE World Congress on Services), Shenzhen, China

8 July

- CSF (IEEE Computer Security Foundations Symposium), Enschede, Netherlands
- EuroS&P (IEEE European Symposium on Security and Privacy), Vienna, Austria

15 July

- CISOSE (IEEE Int'l Congress On Intelligent and Service-Oriented Systems Eng.), Shanghai, China
- ICME (IEEE Int'l Conf. on Multimedia and Expo), Niagara Falls, Canada
- SCC (IEEE Space Computing Conf.), Mountain View, USA
- SMC-IT (IEEE Int'l Conf. on Space Mission Challenges for Information Technology), Mountain View, USA

16 July

- ICDCS (IEEE Int'l Conf. on Distributed Computing Systems), Jersey City, USA

22 July

- ICCP (IEEE Int'l Conf. on Computational Photography), Lausanne, Switzerland

24 July

- ASAP (IEEE Int'l Conf. on Application-specific Systems, Architectures and Processors), Hong Kong

AUGUST

7 August

- IRI (IEEE Int'l Conf. on Information Reuse and Integration for Data Science), San Jose, USA
- MIPR (IEEE Int'l Conf. on Multimedia Information Processing and Retrieval), San Jose, USA

25 Aug

- HCS (IEEE Hot Chips Symposium), Stanford, USA

27 Aug

- SustainTech (IEEE SustainTech Expo: Technology Solutions for a Sustainable Future), San Diego, USA

SEPTEMBER

2 September

- VL/HCC (IEEE Symposium on Visual Languages and Human-Centric Computing), Liverpool, UK

23 September

- MASS (IEEE Int'l Conf. on Mobile Ad-Hoc and Smart Systems), Seoul, South Korea

24 September

- IC2E (2024 IEEE Int'l Conf. on Cloud Eng.), Paphos, Cyprus

Learn more about
IEEE Computer
Society conferences

computer.org/conferences