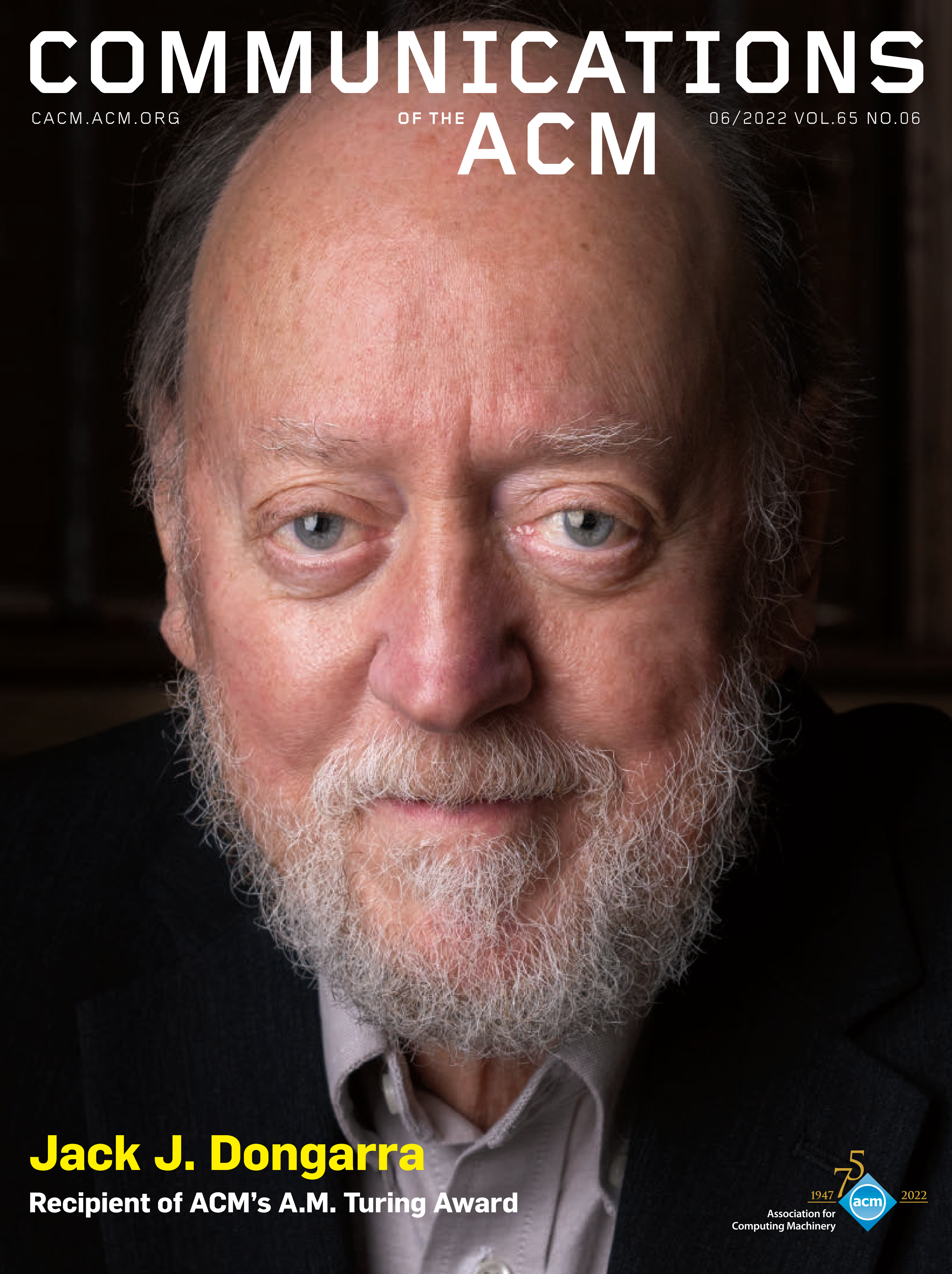
COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

06/2022 VOL.65 NO.06



Jack J. Dongarra

Recipient of ACM's A.M. Turing Award

75 1947 2022

acm

Association for Computing Machinery

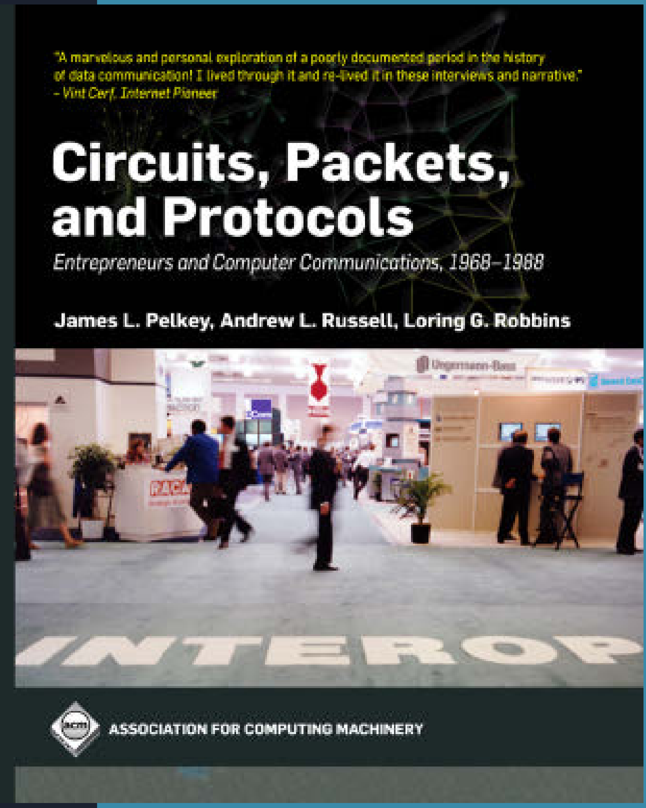

Trust Extension as a Mechanism for Secure Code Execution on Commodity Computers
Bryan Jeffrey Parno
Text Data Management and Analysis
A Practical Introduction to Information Retrieval and Text Mining
ChengXiang Zhai
Sean Massung
The Sparse Fourier Transform
Theory and Practice
Haitham Hassanieh
Frontiers of Multimedia Research
Shih-Fu Chang
Editor
Smarter than Their Machines
Oral Histories of Pioneers in Interactive Computing
John Cullinane
The VR Book
Human-Centered Design for Virtual Reality
Jason Jerald, Ph.D.
Verified Functional Programming in Agda
Aaron Stump
A Framework for Scientific Discovery through Video Games
Seth Cooper
An Architecture for Fast and General Data Processing on Large Clusters
Matei Zaharia
The Handbook of Multimodal-Multisensor Interfaces
Volume 1
Foundations, User Modeling, and Common Modality Combinations

ACM BOOKS

Collection II

As recently as 1968, computer scientists were uncertain how best to interconnect even two computers. The notion that within a few decades the challenge would be how to interconnect millions of computers around the globe was too far-fetched to contemplate. Yet, by 1988, that is precisely what was happening. The products and devices developed in the intervening years—such as modems, multiplexers, local area networks, and routers—became the linchpins of the global digital society. How did such revolutionary innovation occur?

This book tells the story of the entrepreneurs who were able to harness and join two factors: the energy of computer science researchers supported by governments and universities, and the tremendous commercial demand for Internetworking computers. The centerpiece of this history comes from unpublished interviews from the late 1980s with over 80 computing industry pioneers, including Paul Baran, J.C.R. Licklider, Vint Cerf, Robert Kahn, Larry Roberts, and Robert Metcalfe. These individuals give us unique insights into the creation of multi-billion dollar markets for computer-communications equipment, and they reveal how entrepreneurs struggled with failure, uncertainty, and the limits of knowledge.



Circuits, Packets, and Protocols

Entrepreneurs and Computer Communications, 1968-1998

James L. Pelkey
Andrew L. Russell
Loring G. Robbins

ISBN: 978-1-4503-9727-8
DOI: 10.1145/3502372

COMMUNICATIONS OF THE ACM

IMAGE FROM SHUTTERSTOCK.COM

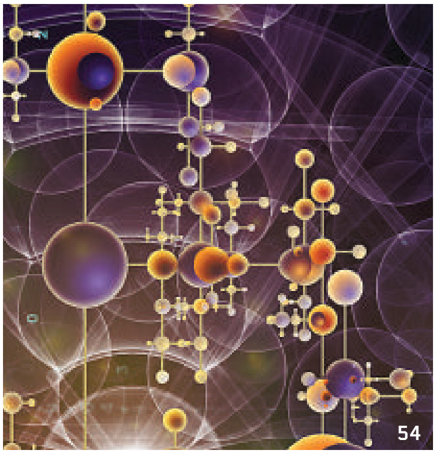
1947 75 2022
Association for Computing Machinery
Advancing Computing as a Science & Profession

06/2022
VOL. 65 NO. 06

Practice

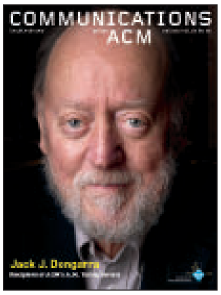
Articles' development led by acmqueue
queue.acm.org

Contributed Articles



Review Articles

Research Highlights



About the Cover:
Jack J. Dongarra, recipient of the 2021 ACM A.M. Turing Award, was photographed on April 8, 2022, in Ayres Hall on the campus of the University of Tennessee, Knoxville. Cover photo by Alexander Berg.

IMAGES BY: (L) SLAWOMIR ZELASKO; (M) ANDREW OSTROVSKY; (R) CREATIVE ICON STYLES

COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

Communications of the ACM is the leading monthly print and online magazine for the computing and information technology fields. *Communications* is recognized as the most trusted and knowledgeable source of industry information for today's computing professional. *Communications* brings its readership in-depth coverage of emerging areas of computer science, new trends in information technology, and practical applications. Industry leaders use *Communications* as a platform to present and debate various technology implications, public policies, engineering challenges, and market trends. The prestige and unmatched reputation that *Communications of the ACM* enjoys today is built upon a 50-year commitment to high-quality editorial content and a steadfast dedication to advancing the arts, sciences, and applications of information technology.

ACM, the world's largest educational and scientific computing society, delivers resources that advance computing as a science and profession. ACM provides the computing field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources.

CONTACT POINTS

Copyright permission
permissions@hq.acm.org
Calendar items
calendar@cacm.acm.org
Change of address
acmhelp@acm.org
Letters to the Editor
letters@cacm.acm.org

WEBSITE

http://cacm.acm.org

AUTHOR GUIDELINES

http://cacm.acm.org/about-communications/author-center

ACM U.S. TECHNOLOGY POLICY OFFICE

Adam Eisgrau
Director of Global Policy and Public Affairs
1701 Pennsylvania Ave NW, Suite 200, Washington, DC 20006 USA
T (202) 580-6555; acmpo@acm.org

COMPUTER SCIENCE TEACHERS ASSOCIATION

Jake Baskin
Executive Director

Association for Computing Machinery (ACM)
1601 Broadway, 10th Floor
New York, NY 10019-7434 USA
T (212) 869-7440; F (212) 869-0481

Subscriptions
An annual subscription cost is included in ACM member dues of $99 ($40 of which is allocated to a subscription to *Communications*); for students, cost is included in $42 dues ($20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is $269.

Single Copies
Single copies of *Communications of the ACM* are available for purchase. Please contact acmhelp@acm.org.

COMMUNICATIONS OF THE ACM

(ISSN 0001-0782) is published monthly by ACM Media, 1601 Broadway, 10th Floor New York, NY 10019-7434 USA. Periodicals postage paid at New York, NY 10001, and other mailing offices.

POSTMASTER

Please send address changes to *Communications of the ACM*
1601 Broadway, 10th Floor
New York, NY 10019-7434 USA

Printed in the USA.

Association for Computing Machinery

PLEASE RECYCLE THIS MAGAZINE

DOI:10.1145/3533672

Andrew A. Chien

Five Years as Editor-in-Chief of *Communications*

This is my last editorial as Editor-in-Chief of *Communications*,[a] so it is a moment to share learnings and, of course, to reflect on accomplishments.

First, we launched the Regional Special Sections (RSS) in November 2018 with a spotlight on computing in the China Region. With 40 pages of articles, spanning tech idols to gaming to computing culture to fintech and "superAI," the first RSS created an excitement that inspired and challenged co-hosts of the Europe, India, East Asia and Oceania, Latin America, and Arabia Regions. In just three years, we have circumnavigated the globe,[b] and with the second Europe Region Section (April 2022) and India Region Section (November 2022), a new circuit is well under way!

The RSS are an exciting read for the ACM community (great job by the co-hosts and authors), delivering news insights and perspectives into how computing is shaping and being shaped around the world. As important, the RSS have grown the *Communications* community—each was created by a multinational workshop and led by co-hosts from the region, helping to expand the diversity of the *Communications*' author community. The eight RSS already published include 350 authors, 18 co-hosts, and hundreds of workshop attendees. They reflect diversity and inclusion as a geographic and cultural imperative! Thanks so much to the co-hosts and authors.

Second, we launched the "Computing Enabled Me To … " feature with the goal of highlighting the remarkable breadth of opportunity enjoyed by computing professionals.[c] Launched in May 2020, we have showcased a remarkable breadth of diverse career paths—from ict4d to edtech to robotic automation to acoustic epidemiology to live coding music and more. This spectrum of role models has both diverse career paths and equally varied backgrounds—ethnicity, gender (50% female), geographic region (Mexico, Japan, U.S., U.K., Australia, Indonesia, Kenya). This feature portrays a rich spectrum of role models and opportunities that a career in computing can provide. We published our 12th feature in the series and are producing half a dozen per year.

We are proud of these new initiatives (and the Digital Initiative[d] has just begun!), but the heart of *Communications* is the continued excellence of its research papers and core features. I am honored to have continued this tradition, and in the past five years, we have published over 350 excellent research papers and more than 2,000 Viewpoints, columns, practice, editorials, highlights, counterpoints, and more. From *Communications*' broad base, I have worked hard to expand our view to reflect the runaway excitement in areas such as AI/machine learning, blockchain, cybersecurity, and data science that are driving the rapid growth of our field.

One privilege of being Editor-in-Chief is to take credit for the work of an amazing team (more than 100 members on the Editorial Board, a dozen co-chairs, and nearly a dozen staff). I would like to offer a heartfelt thanks to all of you for generously sharing your time and expertise. Thanks to Jakob Rehof and Sriram Rajamani, who took over and drove the RSS, and to Mei Kobayashi who created "Computing Enabled Me To … ." I am particularly indebted to Diane Crawford (executive editor) and the *Communications* staff—none of this would be possible without you. Also, thanks to the tireless support of two talented deputies—Morgan Denlow and her predecessor, Lihan Chen—who had the courage to take on an ill-defined role and make it successful.

I plan to stay engaged as a senior editor with an open portfolio, finding new ways to improve *Communications'* community.

I encourage you to bring your passion and expertise to support of new EiC as you have with me.

Thanks!

Andrew A. Chien, EDITOR-IN-CHIEF

Andrew A. Chien is the William Eckhardt Distinguished Service Professor in the Department of Computer Science at the University of Chicago, Director of the CERES Center for Unstoppable Computing, and a Senior Scientist at Argonne National Laboratory.

a Chien, A.A. Today's *Communications of the ACM*, July 2017.

b Chien, A.A. Around the World: (The First Time) with *Communications*' Regional Special Sections, March 2021.

c Chien, A.A. and Kobayashi, M. Launching a New Feature in *Communications*, May 2020.

d Chien, A.A. *Communications*' Digital Initiative and Its First Digital Event, April 2022

DOI:10.1145/3533677

Gabriele Kotsis and Vicki L. Hanson

Our ACM Community

As we celebrate ACM's 75th Anniversary this year, it seems a fitting time to consider the community that ACM has fostered. ACM was founded to support the science

of computing and over the years has continued to do this in many ways. We began as a national organization of 52 members called the Eastern Association for Computing, reflecting the initial local scope of its founding members. "Eastern" was dropped a few months later, with the recognition that the organization could eventually reach far beyond what was initially envisioned. In the early years, ACM was comprised entirely of members and volunteer leaders, with paid staff only being hired later as the organization grew beyond what could be supported solely by the members' efforts.

It is a strength of ACM that it has continued to grow and develop new programs. In the 1960s, Special Interest Groups (SIGs) were started, recognizing that professionals would want to connect not only with the computing profession in general, but also with others interested in their areas of specialization. Today, there are 38 SIGs, giving many a sense of professional community in their areas of greatest focus. In the past two decades, ACM also has grown to include Regional Councils that provide recognition and support for initiatives within their region of the world. ACM's programs have grown to provide services to a large global community: as a publisher and archivist of computing's literature, as a developer of curricular guidelines used worldwide, as a source of impartial technical information for policymakers, and as a channel for multiple programs supporting lifelong professional development.

For us (the authors, and we know for many members like us), involvement in ACM began with a desire to be part of our own community of computing professionals. From that, volunteer activities with Regional Councils, SIGs, and ACM's publishing processes naturally emerged. These varied opportunities directly arose because of the strength and diversity of the whole ACM community.

Local and area specializations have become a core part of our mission. But it is important to keep in mind that *all are part* of the larger ACM. It is easy to forget this and doing so can easily lead to unnecessary divisions. While it is common for individuals to identify primarily, for example, with one of the Regional Councils, or one of the SIGs, we see that some members view their identification with these groups as somehow contrasting with their being part of the whole of ACM. The heart of the matter is what appears to be a sentiment by many that they are not part of ACM.

ACM faces another consequence of our growth: the perception that ACM is a business like any other. It is a hard reality that a large multinational, multipurpose organization cannot exist without a solid legal and financial foundation. Without these, Regional Councils, our nearly 200 conferences, ACM publishing, and a host of other impactful ACM activities could not be pursued. The financial stability of the organization serves as the essential backstop allowing contracts to be signed and commitments honored. It is only through the strength of ACM as a whole that this is possible.

We have heard some within ACM speak of "ACM" as if it were something apart from them. We strongly believe it is those in the total ACM community who make ACM the very special and world-leading organization that it is. It is the dedicated volunteers devoting countless hours to leadership, reviewing, committee work, and service who drive ACM activities. It is the ACM members, and many non-members as well, who participate in the larger community and are served by it. It is the dedicated ACM staff who work with and support volunteer leaders and members in numerous ways. And it is also the nearly four million individuals globally who benefit each year from ACM resources, such as the ACM Digital Library.

ACM's structure does in no way lessen our commitment to serving the global computing community. Rather, it enables it, allowing us to support those driven by a vision of what computing as a discipline can accomplish and what we, being joined together in ACM, can accomplish. We need to keep in mind that ACM and its various groups are not at odds with each other. Rather, we are one, working together to make progress possible for all.

Gabriele Kotsis is the President of ACM.

Vicki L. Hanson is the Chief Executive Officer of ACM. She also served as ACM President 2016-2018.

DOI:10.1145/3534934

Vinton G. Cerf

Digital Synergy

Imagine for a moment a company I will call ACRONYM that maintains warehouses around the world from which customer orders for a variety of products are made. Imagine further that some of the inventory is in digital form (think movies, music, ebooks) in addition to being physical in nature. ACRONYM's access to information in digital form turns out to be enormously enabling. The data is readily processed by sophisticated algorithms. *This column is pure speculation on my part* but inventing the imaginary ACRONYM Company helped me imagine how powerful it can be to have data available that permits fact-based analysis of business performance, projections for future business operations, and understanding of the marketplace the business serves.

ACRONYM gets its orders online, so they are already in digital form, with information ACRONYM needs to fulfill the order. Customers provide all the information requested since, without it, ACRONYM can't or won't place the order. The fact that the data is already in machinable form means ACRONYM can (in no particular order or priority):

1. Track the geocentric demand for specific products, which will inform them of where to place inventory and in what quantity in its many warehouses to reduce delivery time;
2. Track seasonal variations in demand to create powerful forecasting tools so that long-lead items can be ordered for inventory in a timely way;
3. Associate demographics with product orders to further analyze and anticipate demand and inform potential advertising campaigns;
4. Use reinforcement and other machine learning tools to prioritize stock keeping levels in warehouses;
5. For perishables, keep track of expiration dates to know automatically when to pull stock from inventory;
6. For potential contamination, automatically identify inventory to be removed and also to make notifications since order information typically includes email and mobile telephone information associated with particular customer orders;
7. Track delivery success rates, timelines, to detect performance variations that might require attention;
8. Keep track of long-tail ordering behaviors that might lead to concentration of long-tail inventory in central locations to be transferred either directly to customers or to distribution warehouses at need, reminding me of Inter-Library Loan systems;
9. Keep track of suppliers and to cope with supply chain disruption by deliberately exercising multiple sources where possible. Brands and brand-substitution come to mind; and
10. Place digital inventory appropriately in distribution networks to reduce data transfer requirements and latency for customer demand.

I am sure readers have already thought of a much longer list of desirable consequences of maintaining current and historic data associated with products, demand variations, seasonal and regional preferences, and other metrics that drive customer interest. One can even imagine deriving a sense of cultural dynamics and change from this data, not only to recognize fads (Hula hoops!) but also more subtle shifts taking place over longer periods of time. How and what we consume are indicators of societal preferences. Advertisements found in 19th-century *National Geographic* magazine offer a remarkable sketch of life in that time period. One imagines sociologists and anthropologists digging into data in the same way they might dig into an archaeological tell. Just as there are disciplines for real "digs" one can imagine disciplines for "digital digs" that guide the quality of data collection and analysis as more data is uncovered.

This also leads me to think about the provenance of data used for these analytical purposes. Provenance factors into data quality as do measures of data integrity. Maintaining data quality from source to final analysis seems intimately connected to the utility of the results obtained. One can imagine testing for digital data and algorithm quality by making predictions and evaluating their success rates. Certificates and digital signatures over hashes of data are obvious potential tools to ensure a kind of end-to-end integrity of information. To the extent that ACRONYM is willing to share any of its accumulated information, one can imagine it could contribute to a wide range of emergency response preparedness planning exercises as might be conducted by a federal emergency response agency, for example.

I am conscious that I may have just spent my time stating the obvious and your time reading it, but the exercise has only reinforced my belief that quality information has become a highly valuable commodity in our increasingly information rich and dependent world.

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012–2014.

DOI:10.1145/3532630

More On Computing's Divided Future

THANK YOU SO much for addressing the danger posed by the current Chinese government's imperialistic ambitions, using every tool at its disposal, including technology and foreign visitors (January 2022 *Communications* Editor's Letter). Considering how we have come to accept Chinese government atrocities in Tibet, you were at least honest enough to remind us of how Hong Kong is being subjugated, to the detriment of diversity. As reported in the *New York Times* (Jan. 30, 2022): "The Chinese Communist Party has found the best model for controlling people," "honed its powers to track and corral people, backed by upgraded technology … ". "This amounts to a potent techno-authoritarian tool."

I would, however, contest the assertion "The situation is not one-sided" as contrasted with China's "… stark undermining of Hong Kong's basic law, and the violent suppression of its free press … " versus "the U.S. government has taken actions that discourage collaboration with Chinese institutions and researchers. Companies have been put on the entity list, blocking contact, and universities labeled as having military ties, limiting their students' study in the U.S." It is a matter of China's policy, practice, or advocacy of extending power and dominion versus the U.S. defending itself from attack and infiltration.

I do agree that it is a great shame that technological exchanges between countries, industries, universities, and researchers cannot be neutral and that we must guard against spying, theft, and attack. We went through this with Russia during the Cold War, but now the opposition is much more formidable.

Warren Scheinin, Redondo Beach, CA

Editor-in-Chief's response:

Thanks for your response. I raised the topic because thoughtful engagement is essential as too many in our community (the international ACM computing professional community) deny the existence of a growing problem. Or even if they recognize and acknowledge it, remain unwilling to engage to manage it. Only with the engagement of the bulk of our membership will ACM be able to address and surmount this rapidly growing challenge and forge a stable international collaborative community. We need the creativity and passion of the entire community, even those denying an issue now to help us find innovative solutions.

More directly, I believe you are mischaracterizing my comments. I called out actions from both the PRC and U.S. because indeed both governments are taking actions that erode our open, collaborative community.[a] *In no way did I suggest they are equal, nor did I characterize who might be viewed as an aggressor or defender. I leave it to others to engage in such discussions, as my focus is on how all of these actions erode trust, communication, and the ability to collaborate. This is to the detriment of computing science, education, and the computing profession.*

Andrew A. Chien, Editor-in-Chief, *Communications of the ACM*, Chicago, IL

Systems and Methods to Detect Patent Vampires

In his November 2021 Kode Vicious column, "Patent Absurdity," George Neville-Neil advises software developers to avoid reading patents. He also recommends disallowing software patents, suggesting the objective of patents is to serve as weapons to achieve financial benefit and that they are contemptible and useless, proposing interacting with lawyers is like interacting with vampires, and not the friendly kind.

First, there is no such patent type as a "software patent"—this designation does not exist. There are inventions that can be implemented by using software; however, the inventions are processes and methods, instructions that can be implemented to create useful functionality. Second, many inventions give their readers the ability to quickly become exposed to novel concepts that are often easy to understand; for example, a new authentication method that is fundamentally different than all existing techniques, such as standard and biometrics-based passwords. Other inventions quickly expose the reader to novel computational methods; for example, a new feature selection method that competes with widely used techniques, all attempting to achieve better prediction and classification performance. Indeed, patent claims are often written in a legal language that is not easily understood. However, the content and figures are often readable even by nontechnical individuals. Such content is different from scientific manuscripts, which usually require significant expertise to understand in full.

Why does Neville-Neil express such negative opinions about software-related patents? This is likely because of a minority of people who abuse the patenting infrastructure. The U.S. Patent and Trademark Office as well as lawyers (the good kind) have taken a variety of measures to prevent issuing patents for obscure and/or abstract concepts. Furthermore, many of the people involved in patenting are legitimate lawyers and patent agents who truly wish to protect the novel, useful, and valuable inventions of innovators at the company they work for—and there is nothing wrong with that! An invention that describes a novel computational technique may be formed by a group of scientists and engineers working together for hundreds of hours. They and the company they work for deserve to financially benefit from their efforts.

What about this minority of people who abuse the system? Like how inventors form a novel system or method, legislative decision makers need to continue working toward creating more efficient methodologies to detect and then quarantine all patent vampires.

Uri Kartoun, Cambridge, MA

Author's response:

If only it was a minority that abused the system. The stream of cases passing through

a Cracks in open collaboration in universities. *Commun. ACM 63*, 1 (Jan. 2020); Computing's divided future. *Commun. ACM 64*, 1 (Jan. 2021).

one, very famous, court in East Texas seems to show that the problem is large, and with another court in West Texas recently sending its own "welcome to patent trolls," perhaps to compete with East Texas, the problems will only grow; see https://bit.ly/3H0yig3

George Neville-Neil,
Brooklyn, NY

Census Reconsiderations

Garfinkel, Abowd, and Martindale in "Understanding database reconstruction attacks on public data" (*Communications*, March 2019) highlight the dangers of database reconstruction attacks (DRAs). These U.S. Census Bureau methodologists illustrate DRAs using a small dataset of seven records, which they claim to be realistic because "the 2010 U.S. Census contained 1,539,183 census blocks in the 50 states and the District of Columbia with between one and seven residents." They protect the statistical output released for this dataset using cell suppression, more precisely "the rule of three," according to which cells sourced from fewer than three individuals are suppressed. The protected output is shown in Table 1 of their article, where (D) stands for a suppressed cell.

They show the original dataset can be uniquely reconstructed from their Table 1 by using a sophisticated SAT solver—in fact, reconstruction is feasible with much less apparatus: differencing entries in Table 1 and using simple logic and arithmetic reasoning success (see Muralidhar and Domingo-Ferrer, "Database reconstruction is very difficult in practice." UNECE/Eurostat Work Session on Stat. Data Confidentiality 2021.) Based on their analysis, they conclude protection of databases using suppression is inadequate. It turns out the reconstruction of Garfinkel et al. is based entirely on the incorrect application of (primary and complementary) suppression, so this conclusion is completely unwarranted:

- Under the rule of three, in addition to suppressing cells sourced from fewer than three individuals (primary suppressions), cells whose value would allow deriving the primary suppressions via subtraction must also be suppressed (complementary suppressions). See U.S. Census Bureau documents "American Community Survey: Data Suppression" and "Disclosure Avoidance and the Census." This requires cells in line 4A of Table 1 to be suppressed. Failing to suppress 4A allows differencing 2C and 4A to obtain 4B (the values for the only Black American male individual in the dataset).
- Releasing the median Age discloses the age of a single individual (when the group size is odd) or the average age of two individuals (when the group size is even). Hence, medians cannot be released under the rule of three.
- In contrast, there is no need to suppress the count and the mean in line 3A, since they affect three individuals (Single Adults) and do not need to be eliminated as complementary suppressions.

Without 4A, with 3A and without medians, there are thousands of datasets compatible with the remaining counts and means. No unique gender assignment to the seven individuals is possible. Also, uniquely reconstructing Age values without the medians is infeasible, as there are many Age assignments that fit the extant means and counts.

Hence, correct application of cell suppression can perfectly protect the example dataset. What the authors have proven, albeit unintentionally, is that simple disclosure prevention techniques, properly applied, are effective in preventing reconstruction of even very small (hypothetical) datasets created by senior U.S. Census Bureau methodologists to show the very opposite. Database reconstruction may not be so easy after all and this must be taken into account in the ongoing process of protecting the 2020 Census outputs.

Krishnamurty Muralidhar,
Norman, OK, and
Josep Domingo-Ferrer, Catalonia

Authors' response:

We thank Muralidhar and Domingo-Ferrer for their contribution to the example, and we note that the Census Bureau chose to use differential privacy rather than the alternative, massive cell suppression for the 2020 Census.

Simson Garfinkel, Arlington, VA,
John M. Abowd, Washington, D.C., and
Christian Martindale, Durham, NC

Communications welcomes your opinion. To submit a Letter to the Editor, please limit your comments to 500 words or less, and send to letters@cacm.acm.org

Coming Next Month in COMMUNICATIONS

Plus, the latest news about building a practical quantum computer, how brain implants work, and blocking surveillance with makeup.

ACM ON A MISSION TO SOLVE TOMORROW.

Dear Colleague,

Without computing professionals like you, the world might not know the modern operating system, digital cryptography, or smartphone technology to name an obvious few.

For over 70 years, ACM has helped computing professionals be their most creative, connect to peers, and see what's next, and inspired them to advance the profession and make a positive impact.

We believe in constantly redefining what computing can and should do.

ACM offers the resources, access and tools to invent the future. No one has a larger global network of professional peers. No one has more exclusive content. No one presents more forward-looking events. Or confers more prestigious awards. Or provides a more comprehensive learning center.

Here are just some of the ways ACM Membership will support your professional growth and keep you informed of emerging trends and technologies:

- Subscription to ACM's flagship publication ***Communications of the ACM***
- Online books, courses, and videos through the **ACM Learning Center**
- Discounts on registration fees to ACM Special Interest Group conferences
- Subscription savings on specialty magazines and research journals
- The opportunity to subscribe to the **ACM Digital Library**, the world's largest and most respected computing resource

Joining ACM means you dare to be the best computing professional you can be. It means you believe in advancing the computing profession as a force for good. And it means joining your peers in your commitment to solving tomorrow's challenges.

Sincerely,

Gabriele Kotsis
President
Association for Computing Machinery

Association for Computing Machinery

Advancing Computing as a Science & Profession

SHAPE THE FUTURE OF COMPUTING.
JOIN ACM TODAY.

www.acm.org/join/CAPP

SELECT ONE MEMBERSHIP OPTION

ACM PROFESSIONAL MEMBERSHIP:

- ❑ Professional Membership: $99 USD
- ❑ Professional Membership plus ACM Digital Library: $198 USD ($99 dues + $99 DL)

ACM STUDENT MEMBERSHIP:

- ❑ Student Membership: $19 USD
- ❑ Student Membership plus ACM Digital Library: $42 USD
- ❑ Student Membership plus Print *CACM* Magazine: $42 USD
- ❑ Student Membership with ACM Digital Library plus Print *CACM* Magazine: $62 USD

❑ **Join ACM-W:** ACM-W supports, celebrates, and advocates internationally for the full engagement of women in computing. Membership in ACM-W is open to all ACM members and is free of charge.

PAYMENT INFORMATION

Name

Mailing Address

City/State/Province

ZIP/Postal Code/Country

❑ Please do not release my postal address to third parties

Email Address

❑ Yes, please send me ACM Announcements via email

❑ No, please do not send me ACM Announcements via email

❑ AMEX ❑ VISA/MasterCard ❑ Check/money order

Credit Card #

Exp. Date

Signature

Purposes of ACM

ACM is dedicated to:

1) Advancing the art, science, engineering, and application of information technology
2) Fostering the open interchange of information to serve both professionals and the public
3) Promoting the highest professional and ethics standards

By joining ACM, I agree to abide by ACM's Code of Ethics (www.acm.org/code-of-ethics) and ACM's Policy Against Harassment (www.acm.org/about-acm/policy-against-harassment).

I acknowledge ACM's Policy Against Harassment and agree that behavior such as the following will constitute grounds for actions against me:

- Abusive action directed at an individual, such as threats, intimidation, or bullying
- Racism, homophobia, or other behavior that discriminates against a group or class of people
- Sexual harassment of any kind, such as unwelcome sexual advances or words/actions of a sexual nature

BE CREATIVE. STAY CONNECTED. KEEP INVENTING.

Association for Computing Machinery

ACM General Post Office
P.O. Box 30777
New York, NY 10087-0777

1-800-342-6626 (US & Canada)
1-212-626-0500 (Global)
Hours: 8:30AM - 4:30PM (US EST)

Fax: 212-944-1318
acmhelp@acm.org
acm.org/join/CAPP

BLOG@CACM

The *Communications* website, http://cacm.acm.org, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

twitter

Follow us on Twitter at http://twitter.com/blogCACM

DOI:10.1145/3530685 http://cacm.acm.org/blogs/blog-cacm

The Role of Math in IT Education

Andrei Sukhov considers why and how the foundations of teaching mathematics for information technology specialties need to be revised.

Andrei Sukhov
How to Teach Mathematical Disciplines for IT Specialties
January 19, 2022
https://bit.ly/3HMpRoX

The role of mathematical disciplines in IT education is difficult to overestimate.

First of all, mathematics helps to develop algorithmic thinking, as mathematical theories are based on abstract concepts. Mastering any section of fundamental mathematics allows you to operate with abstract concepts to find new patterns.

However, undergraduate IT students often doubt the need to study mathematical disciplines. Their main argument is that the study of mathematics takes a lot of time and effort, but gives very little for the practical application of the acquired knowledge. Therefore, the time spent on studying mathematics can be better used for special courses in IT disciplines.

That is why I decided to discuss the problems of teaching mathematical disciplines, and to propose a number of new approaches.

Note that knowledge of mathematics is particularly in demand at the graduate level. My foreign colleagues often ask me to recommend a programming student to graduate school, but with a mandatory knowledge of mathematics. In Russia, computer science faculties have several areas of study, such as applied mathematics and physics. The curriculum for these specialties contains a considerable portion of programming, and various areas of mathematics and physics. However, the popularity of such areas of training is declining, as is the average level of incoming applicants.

Thus, a revision of the foundations of teaching mathematics for IT specialties is required.

First, we need to have a small number of required courses. Their composition should be discussed, but in my opinion, it is necessary to have only three basic courses. These are differential and integral calculus, basics of algebra, and probability theory with mathematical statistics.

Most of the mathematical theories should be taught as part of new comprehensive courses. Such courses can be devoted to a narrow area of IT technologies and should contain the following three main components:

- basic information from the section on fundamental mathematics;
- formulation of an applied problem and its solution using the studied mathematical approaches, and
- practical implementation of the obtained solutions by means of IT technologies.

The main thing such courses should teach is to go through all the stages from abstract fundamental knowledge to a full-fledged application. Moreover, many applications, particularly the best ones, are based on fundamental knowledge. However, students often have no idea how to get through this path from start to finish. They also do not understand what knowledge is required. At best, they are taught applied mathematics and its application to solve some problems.

In principle, many sections of fundamental mathematics are of applied importance; however, it is quite difficult to develop a full-fledged comprehensive course. This difficulty can be attributed to the large number of consistent conclusions that need to be presented in such a course. Such a presentation requires a broad outlook from the author.

At present, I have begun preparing such a course, using which I will illustrate the main features of this approach. Let us now dwell on the main provisions and content of the course "Traffic model of the backbone network and justification of the threshold method for detecting DDoS (distributed denial-of-service) attacks."

Such a course is fully consistent with the idea of an integrated approach, and contains the following main components:

- queuing theory,
- its applications for describing traffic on the backbone network,
- applied traffic model,
- determination of the abnormal state of the network, and
- substantiation of the threshold value method for identifying sources of DDoS attacks.

Queuing theory is a branch of probability theory based on the problem of the processes of death and reproduction. Advanced research in this area has led to several applications for a wide range of real socioeconomic and demographic processes. However, the first field of application of queuing theory was telecommunications. The main provisions and methods of analysis of the queuing theory in telecommunications are brilliantly presented in textbooks.[1,2] These textbooks are used as a methodological basis for the first component of the course, and should be recommended to students as additional literature.

The next step in building a traffic model on a section of the backbone network was made in the article.[3] By means of the queuing theory, expressions were found for the average traffic on a backbone link and for its variations at short timescales. Note that the generalizations were made at the level of flows, not packets. In particular, the expressions for traffic include the average flow size and the average rate of appearance of new flows on the studied backbone section.

"In my opinion, it is necessary to have only three basic courses: differential and integral calculus, basics of algebra, and probability theory with mathematical statistics."

However, obtaining expressions for traffic and its short-term variation does not yet mean building a full-fledged traffic model. Such a model should define the area of normal operation, as well as highlighting anomalous network states. All these goals can be achieved if the network state is described by two variables: the number of active flows in the network section and the link load in bits per second.[4] Then, the set of network states will be represented by a set of points on the plane, with the abscissa equal to the number of active flows and the ordinate representing the link load.

On this plane, we can build a curve from the averaged values and select a straight part on it corresponding to the operating mode. Depending on the quantile, it is possible to define a parabolic region with a central axis in the form of this straight line. Points corresponding to network states will fall into this area. If several successive states go beyond this area, then we can talk about the anomalous state of the network.

The conducted experiments have shown that the values for some network variables increase many times during DDoS attacks. This fact was first discovered for the number of active flows generated by a single external IP address. Subsequently, it turned out that such variables include incoming TCP and UDP traffic and the number of calls to the Web or proxy server.

In Sukhov, Andrei M., et al.[5], it was proven that for all of these variables, it is possible to find a threshold value. If the threshold is exceeded, we should talk about a DDoS attack. This article describes how to find threshold values and formulates rules for determining the attacking IP addresses. Thus, the theoretical part of the course can be considered to be complete.

In the practical part of the course, it is necessary to create tools to determine the beginning of an attack, such as the IP addresses from which an attack is being carried out, and develop ways to block attacking traffic. The practical implementation of these tools can be carried out on the basis of a number of technologies. These can be Linux utilities, SDN modules, NetFlow, and sFlow collectors.

Students must independently choose a technology for the practical implementation of a theoretical model and justify their choice. It is also necessary to propose a method for restricting traffic from attacking IP addresses, as the theoretical model does not answer this question. The developed software should detect an attack as quickly as possible and start blocking the attacking traffic, and after the attack stops, remove all the restrictions.

I hope to have this course ready by fall 2022 and to offer it as an elective course. In principle, I would very much like to hear comments on the proposed course, as well as on the concept of comprehensive courses in the study of mathematics. In addition, I would like to offer cooperation in the development of topics for comprehensive courses for everyone. Please respond in the comments if you have any suggestions or comments.

Footnotes

1. Kleinrock, L. *Theory, volume 1, Queueing Systems.* (1975).
2. Gnedenko, B.V. and Kovalenko, I.N. *Introduction to Queueing Theory*. Birkhauser Boston Inc., 1989.
3. Barakat, C. et al. Modeling Internet backbone traffic at the flow level. *IEEE Transactions on Signal Processing* 51.8 (2003), 2111–2124.
4. Sukhov, A.M. et al. Active flows in diagnostic of troubleshooting on backbone links. *Journal of High Speed Networks* 18.1 (2011), 69–81.
5. Sukhov, A.M., Sagatov, E.S., and Baskakov, A.V. Rank distribution for determining the threshold values of network variables and the analysis of DDoS attacks. *Procedia engineering* 201 (2017), 417–427.

Andrei Sukhov (asukhov@acm.org) is a professor of HSE University and a senior member of ACM.

Profile | DOI:10.1145/3530689

Neil Savage

Always Improving Performance

Jack J. Dongarra is the recipient of the 2021 ACM A.M. Turing Award for his pioneering contributions to numerical algorithms and libraries that enabled high-performance computational software to keep pace with exponential hardware improvements for over four decades.

AS A YOUNG MAN, Jack Dongarra thought he would probably teach science to high school students. That was his plan when he enrolled at Chicago State College, which had become Chicago State University by the time he graduated in 1972. Over the course of his studies, he began to be fascinated by computers. In his senior year, physics professor Harvey Leff suggested he apply for an internship at nearby Argonne National Laboratory, where he could gain some computing experience.

There, Dongarra joined a group developing EISPACK, a software library for calculating eigenvalues, components of linear algebra that are important to performing simulations of chemistry and physics. It was a heady experience. "I wasn't really a terrific, outstanding student," Dongarra recalls. "I was thrown into a group of 40 or 50 people from around the country who came from top universities and I got to mix with them." Project leader Brian Smith became his mentor. "He was very, very patient with me. I didn't have a very extensive background in computing, and he gave me attention and guided me along."

The experience changed his plans. After earning his degree in mathematics, he began a master's program in computer science at the Illinois Institute of Technology. This was the beginning of a career in which he helped usher in high-performance computing by creating software libraries that allowed programs to run on various processors. It was for that work that Dongarra has been named recipient of the 2021 ACM A.M. Turing Award.

He continued to work at Argonne one day a week while in graduate school and, once he graduated, took a full-time job at the laboratory, where he continued to work on EISPACK. The software was intended to be portable, so it could run on different machines. "We sort of expect that to happen today as a matter of course," he says, "but in those days, it wasn't so easy to do."

Back then, there was no standardization among computers. Today, the standard known as IEEE Arithmetic defines how numbers are handled by computers, but in the 1970s, a machine from IBM would not use the same amount of bits to represent a number as did a machine from Control Data Corporation, and a UNIVAC computer would be different from both. EISPACK had to be designed to work across those machines with only minor changes.

Dongarra followed that project with LINPACK, a software library for linear algebra, designed to solve systems of equations. "What we do in LINPACK and EISPACK is really the basis for much of scientific computing," says Cleve Moler, then a colleague on the project and a professor at the University of New Mexico (UNM), who would later go on to found the computing software company MathWorks. Moler convinced Dongarra to come to New Mexico and study with him. Dongarra took a leave of absence from Argonne and moved to Albuquerque, where in 1980 he earned a Ph.D. in applied mathematics from UNM. While working on his doctorate, he also worked at Los Alamos National Laboratory, where the first Cray supercomputer had been installed, presenting computer scientists with the challenge of making algorithms run on its novel architecture. Using a test program that came to be known as the LINPACK

PHOTO BY ALEXANDER BERG

benchmark, Dongarra discovered a timing error in the Cray that was causing it to give the wrong answer.

In 1989, Dongarra was offered a joint position at the University of Tennessee and Oak Ridge National Laboratory. He accepted, and remains there today. The move allowed him to do some teaching, and being in academia lets him be entrepreneurial, he says, in a way that a national laboratory, with its defined projects, did not.

Dongarra has been successful at what he does thanks to both his intelligence and personality, says Moler, a longtime friend. "It's a beautiful marriage of scientific competence and a kind of humility," Moler says. "He doesn't have any hidden agenda. He's not out to prove himself. He just marches on, old Jack."

Over the course of his career, Dongarra has been involved in the creation of many libraries. LAPACK, for instance, combined LINPACK and EISPACK into a unified package. Another, BLAS, for Basic Linear Algebra Subprograms, was named by the journal *Nature* last year as one of 10 computer codes that transformed science. Dongarra chuckles at that designation. "I'm not sure it's quite as they made it out, but I'm willing to take that," he says. BLAS are "sort of the computational kernels, if you will, the fundamental building blocks of these other libraries."

High Performance, High Standards

BLAS eventually became a de facto standard, thanks to the work of many, Dongarra says. "A group of people in the community got together and said, 'What's the best way to do these things?' We argued, fought, drank beer together, and ultimately came up with a package and made it available to the community, and then had further input on how to refine it before it was cast in stone."

Three qualities have always been important in software libraries he designed, says Dongarra. One is that they can become standard. The second is that they should be portable and able to work on different machines with different architectures, including single processors, parallel computers, multicore nodes and, most recently, nodes containing multiple graphics processing units.

The third quality is that they must run efficiently, which is not always easy to achieve when computer hard-

"It's a beautiful marriage of scientific competence and a kind of humility. He doesn't have any hidden agenda. He's not out to prove himself."

ware keeps evolving. "Every few years, the hardware changes, and if you don't make changes to the software to accommodate for those hardware changes, your software will become inefficient," he says. "We're always sort of in a catch-up game, trying to redesign the software to match the architectural features."

One such challenge arose in the 1990s, with the growth of parallel computing. Originally, computing took place on a single processor, which performed operations sequentially. Later came parallel processors that shared memory. Those gave way to distributed parallel processors, each of which had its own memory. That raised the question of how to pass messages between processors, and each computer company answered it differently. "From a standpoint of writing software that was going to be used by other people, that was going to be a disaster," says Dongarra, who solved the problem by creating the Message Passing Interface with an international group of collaborators.

Another way he dealt with hardware differences was through auto-tuning. Developed in his Automatically Tuned Linear Algebra Software library project, auto-tuning probes different chip designs to discover their basic features, such as how much memory cache they have. It then uses machine learning to create thousands of versions of a program, each with slight variations, and runs all of them on each architecture, to find out which is the most efficient.

In his unending quest for efficiency, Dongarra developed batch computation, which takes the calculation of large matrices—used in simulation and data analysis—and breaks them into smaller blocks that can be solved among different processors.

He also developed mixed-precision arithmetic. The standard way of performing numerical computations has been to use 64-bit floating point arithmetic, which produces results with high accuracy. However, in the growing area of artificial intelligence, that sort of accuracy is not always required, and some work can be done with 16-bit precision and completed in about a quarter of the time. Mixed-precision arithmetic helps programmers figure out which parts of their work needs 64-bit accuracy and which can be done in only 16 bits, rendering the whole system more efficient.

Working with colleagues, Dongarra created the Top500, a list of the world's 500 most powerful supercomputers, ranked by their performance on the LINPACK benchmark. The ranking comes out twice a year, and he predicts the June list will include the first exascale computer, which can perform at least one quintillion (billion-billion) calculations per second. In anticipation of that, he has been working with colleagues around the world to develop a roadmap of what software for such powerful machines should look like.

The Turing Award comes with a $1-million cash prize, and Dongarra says he is not sure what he will do with that. He is still wrapping his head around the honor. "It's an overwhelming situation," he says. "These guys who have won this award are leaders in the field. I've got their books on my bookshelf, I've read their papers, used their techniques. It's incredible. I must give credit to the generations of colleagues, students, and staff whose work and ideas influenced me over the years and I hope I can live up to all the greatness that the Turing Award has recognized and become a role model, as many of the recipients have been, to the next generation of computer scientists."

Neil Savage is a science and technology writer based in Lowell, MA, USA.

Watch Dongarra discuss his work in the exclusive *Communications* video. https://cacm.acm.org/videos/2021-acm-turing-award

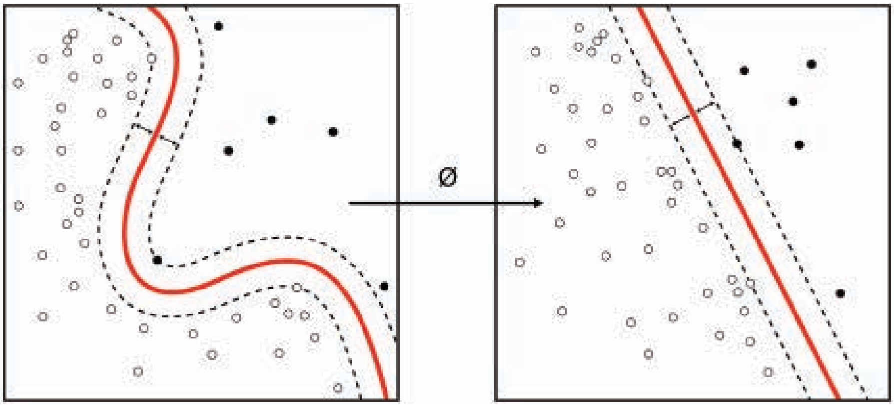
Science | DOI:10.1145/3530686 Don Monroe

A Deeper Understanding of Deep Learning

Kernel methods clarify why neural networks generalize so well.

DEEP LEARNING SHOULD not work as well as it seems to: according to traditional statistics and machine learning, any analysis that has too many adjustable parameters will overfit noisy training data, and then fail when faced with novel test data. In clear violation of this principle, modern neural networks often use vastly more parameters than data points, but they nonetheless generalize to new data quite well.

The shaky theoretical basis for generalization has been noted for many years. One proposal was that neural networks implicitly perform some sort of regularization—a statistical tool that penalizes the use of extra parameters. Yet efforts to formally characterize such an "implicit bias" toward smoother solutions have failed, said Roi Livni, an advanced lecturer in the department of electrical engineering of Israel's Tel Aviv University. "It might be that it's like a needle in a haystack, and if we look further, in the end we will find it. But it also might be that the needle is not there."



Kernel machines like the one above are used to compute non-linearly separable functions into a higher-dimension linearly separable function.

A Profusion of Parameters

Recent research has clarified that learning systems operate in an entirely different regime when they are highly overparameterized, such that more parameters let them generalize *better*. Moreover, this property is shared not just by neural networks but by more comprehensible methods, which makes more systematic analysis possible.

"People were kind of aware that there were two regimes," said Mikhail Belkin, a professor in the Halıcıoğlu Data Science Institute of the University of California, San Diego. However, "I think the clean separation definitely was not understood" prior to work he and colleagues published in 2019. "What you do in practice," such as forced regularization or early stopping of training, "mixes them up."

Belkin and his co-authors systematically increased the complexity of several models and confirmed the classical degradation of generalization. Their analysis revealed a sharp peak in the prediction error as the number of model parameters became high enough to fit every training point exactly. Beyond this threshold, however, generalization improved again, so the overall curve showed what they called "double descent."

A highly overparameterized model—beyond the peak—has a huge, complex manifold of solutions in parameter space that can fit the training data equally well—in fact perfectly, explained Andrea Montanari, Robert and Barbara Kleist Professor in the School of Engineering and professor in the departments of electrical engineering, statistics, and mathematics of Stanford University. Training, which typically starts with a random set of parameters and then repeatedly tweaks them to better match training data, will settle on solutions within this manifold close to the initialization point. "Somehow these have the property, a special simplicity, that makes them generalize well," he said. "This depends on the initialization."

Quantitative metrics of generalization are challenging, though, cautions Gintare Karolina Dziugaite of Google Brain in Toronto, and there are limits on what we should expect from "explanations" for it. One obvious measure is the performance of a trained model when faced with held-out data. "It will be quite precise, but from the explanation perspective, it is essentially silent," she said. General theories, by contrast, do not depend on the details of the data, but "it's well-appreciated at this point that such theories cannot explain deep learning in practice." Dziugaite said. "Any satisfactory theory of generalization should lie between those two regimes."

Dziugaite also noted that memorizing the training set, as overfitting does, could actually be useful in some situations, such as when a dataset includes small subpopulations. A tool that seems to generalize well on average might miss underrepresented examples, such as dark-skinned people in facial recognition data.

Boaz Barak, a professor of computer science at Harvard University, regards generalization as only one aspect of the power of neural networks. "If you want to talk about generalization in a mathematically well-defined way, you need to think of the situation where you have some distribution over the population and you're getting samples from that distribution," he said. "That's just not how things work" for real-world datasets.

Good generalization, on average, also does not address the "fragility" problem, in which neural networks sometimes make inexplicable, egregious errors in response to novel inputs. However, "We are still far from having a way to fix that problem in a principled way," said Montanari.

Kernel Machines

Belkin's "most important discovery was that [the overparameterized regime] is really general," Montanari said. "It's not limited to neural networks." As a result, "people started looking at this phenomenon in simpler models."

Belkin, for example, has championed the venerable kernel machines for both their explanatory and practical power. When used as binary classifiers, kernel machines search in a very high-dimensional feature space for simple surfaces that separate two groups of data points that are intermingled when they are projected into fewer dimensions. To perform this separation, they exploit a mathematical "kernel trick" that computes the distances between pairs of points in the high-dimensional space without the need to compute their actual coordinates.

Kernel machines include support vector machines, which were widely explored for machine learning before the recent ascendence of deep learning. "It's in a sense a simpler model," Belkin said. "If you cannot even understand what's going on with them, then you cannot understand neural networks."

Furthermore, Belkin has come to believe that kernel machines may already contain the most important features of deep learning. "I don't want to say everything about neural networks can be explained by kernels," he said, "but I think that maybe the interesting things about neural networks are representable by kernels now."

In some limiting cases, the connection can be made mathematically precise. One important limit is when a neural network has layers of infinite "width" (as contrasted with the "depth"—the number of layers—that gives deep learning its name). It has long been known that such wide networks, when randomly initialized, can be described as a Gaussian process, which is a type of kernel.

The connection persists during training, as shown in a highly cited 2018 NeurIPS presentation by Arthur Jacot, a graduate student at Switzerland's École Polytechnique Fédérale de Lausanne, and his colleagues. "We approximate the nonlinear model of neural networks by a local linear model," he said. This Neural Tangent Kernel, or NTK, determines precisely how the solution evolves during training.

For infinitely wide networks, the authors showed the NTK does not depend on the training data and does not change during training. Jacot said they are still examining other conditions for a neural network to be in this "NTK regime," including having a large variance in the initial parameters.

"I became more committed to the kernel thing after this NTK paper," said Belkin, "because they essentially showed that wide neural networks are just kernels," which makes generalization easier to model.

Feature Learning

Kernels do not automatically do everything, though. "The main difference between kernel machines and neural networks is that neural networks learn the features from the data," said Barak. "Learning from the data is an important feature of the success of deep learning, so in that sense, if you need to explain it, you need to go beyond kernels." Optimizing feature recognition might even push neural network designers to avoid the NTK regime, he suggested, "because otherwise they may degenerate into kernels."

"It's very easy to come up with examples in which neural networks work well and no kernel methods work well," said Montanari. He suspects the practical success of neural networks is "probably due to a mixture" of the linear part, which is embodied in kernels, with feature learning, which is not.

For his part, Belkin remains hopeful—although not certain—that kernels will be able to do it all, including feature identification. "There are mathematical results showing that neural networks can compute certain things that kernels cannot, he said, but "that doesn't actually show me that real neural networks in practice compute those things."

"It's not always true that neural networks are close to kernel methods," Jacot acknowledged. Still, he emphasized that the NTK can still be defined and describe network evolution even outside of the NTK regime, which makes it easier to analyze what the networks are doing. "With NTK you can really compare different architectures" to see whether they are sensitive to particular features," he said. "That's already very important information."

Convolutional neural networks have proven powerful in image recognition, for example, in part because their internal connections make them insensitive to displacement of an object. "Even though these are not learned features, they are still quite complex, and result from the architecture of the network," Jacot said. "Just having these kind of features leads to a huge improvement in performance" when they built into kernel methods."

For other tasks, though, the features that neural networks identify may be difficult for designers to recognize. For such tasks, Barak suggested, one approach "would be kind of a merging of neural networks and kernels, in the sense that there is the right kernel for the data, and neural networks happen to be a good algorithm to successfully learn that kernel." In addition, "We have some evidence for universal features that depend on the data, not on any particular algorithm that you're using to learn it. If we had a better understanding of that, then maybe generalization would come out the side from that."

Further Reading

Belkin, M., Hsu, D., Ma, S., and Mandal, S.
Reconciling modern machine-learning practice and the classical bias–variance trade-off, *Proc. Nat. Acad. Sci. 116*, 15849 (2019), https://bit.ly/3EgkBYb

Belkin, M.
Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation, *Acta Numerica, 30*, 203 (2021), https://bit.ly/3GUJvhq.

Jacot, A., Gabriel, F., and Hongler, C.
Neural Tangent Kernel: Convergence and Generalization in Neural Networks, 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), https://bit.ly/32bQmo0

Don Monroe is a science and technology writer based in Boston, MA, USA.

Society | DOI:10.1145/3530687

Logan Kugler

Addressing Labor Shortages with Automation

Labor shortages have many companies turning to automation technology, but with mixed outcomes.

U.S. EMPLOYMENT STATISTICS hit a new milestone last year, but not a positive one.

In August 2021, almost 4.3 million workers quit their jobs, according to the U.S. Department of Labor. That's the highest number since the department began tracking voluntary resignations. Their reasons for leaving their jobs vary—the numbers track people who quit for a different position, as well as those who quit without having another job lined up.

While the reasons for quitting vary, one thing is clear: Businesses are having a tough time getting employees to come back. A full 80% of companies surveyed by the Conference Board, a business research nonprofit, say they are now finding it difficult to hire qualified workers.

This is a win for worker wages. Additional Conference Board research predicts U.S. wage costs for companies will rise 3.9% in 2022, the highest jump since 2008. In the U.K., the National Institute of Economic and Social Research expects average weekly earnings growth to jump 5.9% in 2021, compared with 1.8% in 2020.

However, the growth in wages may not last.

The labor shortage, combined with wages increasing to try to attract new employees and keep those they have, have led many businesses to accelerate plans to adopt automation technologies like robots and smarter software. Such adoption is happening across industries like manufacturing, the service industry, and administrative work. It is unclear if this automation will alleviate a permanent shortage of workers who no longer want this type of work—or if it will permanently eliminate jobs that may be in future demand.

PHOTO BY ES SARAWUTH/SHUTTERSTOCK

The recent surge in demand for automation started out as a temporary necessity, says Gad Levanon, vice president of Labor Markets at the Conference Board and founder of the organization's Labor Market Institute.

Many companies had to limit human interactions in order to comply with public health regulations during the pandemic. Using automation to take services online or rollout self-service options was the only way for some companies to keep doing business legally. However, as businesses reopen, many now see automation as a sensible permanent solution to the labor shortfall they now face.

"After massive layoffs during the early months of the pandemic, some have learned to operate with fewer workers by using more automation and other process improvements," says the Conference Board's Levanon says. "2021's severe labor shortage and accelerating wages may incentivize other employers to do the same."

Greater Demand for Fewer People

According to the Association for Advancing Automation (A3), which is described on its website as "the leading global automation trade association of the robotics, machine vision, motion control, and industrial AI industries," orders for robots in North America rose 67% in the second quarter of 2021, compared to the same quarter of the previous year. In fact, it was one of the largest quarters for robot sales on record.

The customers for those industrial robots are in some surprising industries. More than half of Q2 2021 orders came from outside the automotive industry, says A3.

Automotive manufacturing is the leading consumer of industrial robotics, so it is no surprise when the car industry buys robots. The greatest non-automotive industry increases in robotic acquisitions came from sectors like the metals industries, semiconductors and electronics, plastics and rubber, food and consumer goods, and the life sciences, pharmaceutical, and biomed sectors. These numbers reflect the demand for more automation in the manufacturing sector as a whole.

It is a trend Tom Kelly knows all too well. Kelly is the CEO of Automation Alley, a nonprofit that facilitates manufacturing private and public partnerships in Michigan. He says labor shortages have hit manufacturing hard, due to the pandemic and older workers exiting the workforce. During the pandemic, Automation Alley started connecting local manufacturers with automation technologies such as robotics. He and his team are finding manufacturers more willing than ever to test it out.

"We are seeing a demand for the lease of [automation] technologies," Kelly says. That is because it permits companies to try out the automation equipment without first buying it outright, he says.

Kelly is not worried that automation will gut employment in the manufacturing sector. Not only does it increase productivity, but it can also empower workers to move up the value chain, he says. That allows them

"Most factory workers would welcome automation to free up their time to get upskilled to do more meaningful work."

to do more valuable work with higher output, which can positively impact their wages.

"Automation reduces time spent on repetitive tasks, and most factory workers would welcome automation to free up their time to get upskilled to do more meaningful work, or work that involves creativity and strategy," Kelly says. "This won't replace people, but will increase flexibility for workers."

However, automation doesn't always empower workers in other sectors; in some sectors, automation is replacing humans entirely.

A range of companies in food service are experimenting with advanced automation that could reduce headcount permanently. As the industry hit hardest by the pandemic and the resulting labor shortage, food service players are desperate for automated solutions.

Fast food chain White Castle is rolling out a robot named Flippy built by Miso Robotics. Flippy can automatically cook different foods. The latest version of the robot can man an entire fry cooking station by itself, without human intervention.

KFC Korea is also going heavy on automation. The company recently partnered with Hyundai Robotics to automate the chicken-frying process. The work hinges on using collaborative robots within the kitchen to churn out consistent-quality chicken at speed.

Automation is not just eliminating human labor that makes food; it also is doing the work of food delivery labor. The food delivery market has tripled in size since 2017 to more than $150 billion, according to data from market research firm McKinsey, which found the market has doubled since the beginning of the pandemic. To handle that burgeoning demand amid a labor shortage, companies want to automate their delivery vehicles.

Pizza chain Domino's has partnered with a company called Nuro to test self-driving pizza delivery robots. After placing an order, customers meet the small robotic vehicle at their doorstep, where they can enter a

In Remembrance

Andrea Pohoreckyj Danyluk 1963–2022

The ACM Education Board is deeply saddened by the loss of long-time ACM volunteer and Education Board member Andrea Pohoreckyj Danyluk, who passed away Thursday, March 3, 2022, at age 59 after a valiant battle with pancreatic cancer.

Andrea completed her bachelor's degree in mathematics and computer science at Vassar College in 1984 and her Ph.D. in Computer Science from Columbia University in 1991. After working in the research department at Nynex until 1994, Andrea became the first female computer science professor at Williams College where she taught for 27 years, ultimately serving as the Mary A. and William Wirt Warren Professor of Computer Science, Emerita. Throughout her teaching career, Andrea was especially known for her passion for her discipline, her wonderful sense of humor, and her deep commitment to mentoring women students and colleagues.

Andrea's deep and lasting contribution to computer science education is evidenced by extensive volunteer work for many organizations including ACM. She served on the Steering Committee for the Computer Science Curriculum 2013 and most recently as the co-chair (along with Paul Leidig) of the ACM Data Science initiative that released the *Computing Competencies for Undergraduate Data Science Curricula* report in 2021.

Andrea profoundly impacted the lives of her family, friends, students, and colleagues. Her ACM friends will forever remember Andrea as a passionate educator and mentor with a warm spirit, boundless energy, and infectious laughter. She brought intelligence and joy to every ACM project on which she served.

Andrea is survived by her children Katya and Stephan and her husband Andrew Danyluk. ACM offers its deepest condolences to her family, friends, colleagues and the many former students whose lives she touched. She is deeply missed by all her friends at ACM.

For more information, please visit https://bit.ly/3xUpsOU.

—The ACM Education Board

custom PIN into a door on the vehicle and retrieve their food.

Grubhub has started to experiment with automating food delivery, too. The company has partnered with Yandex NV to test self-driving delivery vehicles on college campuses. Other major fast food chains, including Chick-fil-A, have similar autonomous delivery initiatives in the works.

In fact, the pursuit of automation is so fervent in this industry that an entire new category of automation called food tech has arisen. Food tech describes the market for robotics, automation, and data-driven technology in the food industry, which includes both manufacturers of food products, and the logistics sites and apps that deliver them to consumers. Driven by the pandemic and the continuing labor shortage, this category is expected to explode to a $342-billion-per-year market by 2027, according to Emergen Research.

It is not just food service and food tech that are pursuing automation, either. Administrative workers are also under increased threat of replacement by automation, says Levanon. The need to increase contactless payments and transactions during the pandemic, as well as temporary or permanent office closings, has furthered automation in this space. At the same time, the need for in-person administrative or customer-facing staff was reduced or eliminated entirely.

"The accelerated digital transformation of both business and consumer activities now makes it easier to eliminate routine jobs," says Levanon. "This resulted in many in-person customer service positions, such as health care receptionists, commercial banking tellers, and reservation ticket agents being eliminated in favor of online help and automation."

The pursuit of automation is so fervent in the food sector that an entire new category of automation called food tech has arisen.

The Automated Future

There is no doubt automation is replacing certain job functions, but will automation actually deprive human laborers of employment, or just fill jobs or job functions nobody wants? For clues, look to an industry that already has embraced automation to solve labor shortages: agriculture.

"In agriculture, harvesters were developed for field crops like wheat and soy many years ago," says Diane Charlton, a professor who studies farm labor at Montana State University. "Firms are incentivized to automate because fewer workers are willing to work in agriculture."

The agricultural labor market has long since had built-in friction that mirrors our current labor shortage. Even if farmers offer higher wages, there still may not be enough workers interested or available to fill positions. That's because immigrant labor is the backbone of agricultural labor market, but the flow of this migrant labor each season is not fixed due to a range of political and economic factors. (Not to mention, the work can be back-breaking.)

Consequently, agriculture was forced to automate. Today, automated harvesters do some agricultural work, while other types of robots augment the efforts of human laborers. As a result, the number of workers needed in the field has decreased, says Charlton. However, more workers ended up needed in related agricultural industries, creating different types of jobs. These include jobs managing and maintaining robots, positions demanding more complex skills and commanding higher wages.

"Often, the jobs created are better paying or more comfortable," she says.

However, that future is not certain, notes Charlton. Many individual people and companies were ruined by automation in agriculture, such as small farms that could not afford to expand quickly enough to compete with major agricultural concerns deploying automation at scale. In addition, the skills required for a more automated world are different from those that were rewarded pre-automation. Workers who cannot adapt—or who try to return to jobs in automated industries—may find themselves out of luck.

"As with most major changes to the economy, there are winners and losers," Charlton says.

Further Reading

Ahuja, K.
Ordering in: The rapid evolution of food delivery, *McKinsey & Company*, September 22, 2021, https://mck.co/3vzZptx

Broady, K. et al.
Workers must use their newfound leverage to protect their careers from automation, *Brookings*, December 10, 2021, https://brook.gs/3vxtsSP

Casselman, B.
Workers quitting their jobs hit a record in the U.S. in August, *The New York Times*, Oct. 12, 2021, https://www.nytimes.com/2021/10/12/business/economy/workers-quitting-august.html

Food Tech Market by Technology Type, *Emergen Research*, January 2021, https://www.emergenresearch.com/industry-report/food-tech-market

Levanon, G.
2022 Salary Increase Budgets Are the Highest Since 2008, *The Conference Board*, Dec. 7, 2021, https://www.conference-board.org/blog/labor-markets/2022-salary-increase-budgets

Riaz, S.
UK's record high job vacancies lead to pay rises for new recruits, *Yahoo News*, Dec. 14, 2021, https://news.yahoo.com/u-ks-record-high-job-vacancies-lead-to-pay-rises-for-new-recruits-160818058.html

Robot Orders Increase 67% in Q2 2021 Over Same Period in 2020, Showing Return to Pre-Pandemic Demand for Automation, *Automation.com, Oct. 19, 2021*, https://www.automation.com/en-us/articles/october-2021/robot-orders-increase-67-percent-q2-2021-demand

Smart, T.
Overwhelming Majority of Businesses Report Difficulty Hiring Workers and Retaining Existing Employees, *U.S. News & World Report*, Jun. 2, 2021, https://bit.ly/3vwI4lx

Logan Kugler is a freelance technology writer based in Tampa, FL, USA. He is a regular contributor to *Communications* and has written for nearly 100 major publications.

Technology | DOI:10.1145/3530688

Samuel Greengard

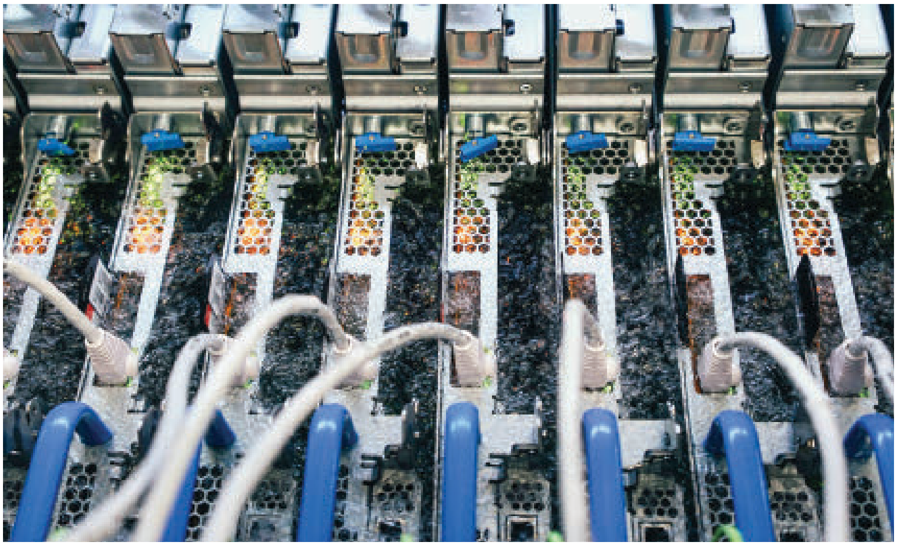
Immersion Cooling Heats Up

Depending on climate conditions, the availability of renewables and other factors, immersion cooling can make a profound difference in both energy consumption and costs.

DATACENTERS ARE BOTH the heroes and villains of the digital age. On one hand, these facilities power the technologies that increasingly run our world. On the other hand, datacenters devour enormous amounts of energy and face growing opposition in many communities. As the world careens deeper into climate crises, finding ways to reduce power consumption is nothing less than critical.

One idea that has lurked in the background for decades is using special fluids to directly or indirectly cool computing devices and other electronic systems. While the concept may seem shocking, dielectric liquids and immersion cooling methods have come of age. These technologies, which already are making inroads in high-performance computing and cryptocurrency mining circles, bathe electronic components in a dielectric (nonconductive) liquid or coolant with strong insulation properties. The circulation of the fluid draws off heat as it comes into contact with the electronics.

Immersion cooling is poised to make a major impact on datacenters. Although IBM and Cray experimented with the technology during the 1960s through the 1980s, advances in design, engineering and fluids are finally making the technology viable and more affordable. "Immersion cooling has advanced remarkably over the last few years," states Lucas Beran, principal analyst at market research and consulting firm Dell'Oro Group. "As the pressure mounts to control power consumption and heat generation in datacenters, the technology has an important role to play."



Circulated liquid carries away the heat generated by servers in this Microsoft datacenter.

Feeling the Heat

As businesses, government entities, and others look for ways to reduce their carbon footprints and trim energy costs, it is clear that conventional air cooling, and even other fluid-based technologies such as Direct Liquid Cooling (DLC), which can be applied to CPUs and GPUs using special plates, are not up to the task.

It also is becoming difficult to squeeze out energy efficiency gains in datacenters by refreshing legacy servers, optimizing data, virtualizing workloads, and turning to green hosting. Thermal management now consumes 30% to 40% of annual energy consumption in datacenters, according to Dell'Oro Group.[a] In fact, datacenters are forecast to consume 8% of the world's electricity by 2030, up from about 1% in 2018.[b]

Energy consumption is spiking in datacenters because per-socket demand for power is increasing rapidly, says Jim Rogers, computing and facilities director for the National Center for Computational Science at Oak Ridge National Laboratory in Tennessee. "Just a few years ago, a CPU might consistently consume about 100 watts. Today's CPUs and GPUs are consistently using more than twice that amount," he says. As the number of computers in a single rack has grown, density has also become an issue. "The net result is that the aggregate or total amount of heat that must be managed has steadily climbed to 30, 40, or 50 kilowatts in a single rack," he says.

These two factors together are pushing conventional cooling systems to their limits. "When manufacturers move above about 15 kilowatts per rack, the strategy to eject the waste heat to the datacenter space by just moving air with server fans effectively ends," Rogers explains. At that point, rear-door heat exchangers, cold plates, and other technologies are necessary. What makes immersion cooling stand out is that the heat generated by various components in the server is directly absorbed by a fluid, which is far more efficient than air at dissipating heat.

a https://www.delloro.com/is-immersion-cooling-the-answer-to-data-center-sustainability/
b https://www.nature.com/articles/d41586-018-06610-y

As heat densities grow, the challenges mount. "Blowing air across components is inefficient and removing heat from air is mechanically expensive," Rogers says. Direct Liquid Cooling (DLC), which also goes by the name Direct-to-Chip, at best can extract only 70% of the heat.[c] Meanwhile, water-based cooling systems, a subset of DLC, are used broadly, but they utilize cold plates circulating water near hot components and must rely on limited thermal transfer. "Immersion cooling solves that specific issue by offering great specific heat absorption capability using non-conductive fluids," according to Rogers.

The real-world impact is significant. The Uptime Institute reports the average datacenter's power usage effectiveness (PUE) rating in 2020 was 1.58—and the figure has been stagnant since 2013.[d] Highly efficient air-based cooling typically delivers a PUE of about 1.2 to 1.4 (the lower figure is more common in cloud datacenters like AWS, while the latter figure is more representative of an enterprise datacenter), but immersion cooling methods lower that figure to approximately 1.03 or better, Beran points out. Depending on climate conditions, the availability of renewables and other factors, immersion cooling can make a profound difference in both energy consumption and costs. The end-game is to get PUE ratings as close as possible to 1.0.

However, PUE, which is the de facto standard for defining energy efficiency in datacenters, does not entirely capture the significance of more advanced liquid and immersion cooling methods because there is a ripple effect, including reducing dependence on fans and other electrical components. Jacqueline Davis, a research analyst at the Uptime Institute, points out that liquid cooling and immersion techniques "profoundly change the profile of datacenter energy consumption."[e]

A More Fluid Approach

Two primary types of immersion cooling exist. For now, the most widely used immersion cooling technique is Single Phase Immersion, which relies on an accessible enclosure filled with dielectric fluid. A pump circulates the dielectric fluid or deionized water in the enclosed space until it comes into contact with a heat exchanger, which pulls the heat out and transfers it to a water circuit, before returning the cooler fluid back to the enclosure. With Single Phase Immersion, the coolant never boils or freezes, and there is little or no risk of evaporation. Typically, servers are installed vertically inside a horizontally oriented cooling bath.

Depending on climate conditions, the availability of renewables and other factors, immersion cooling can have a profound impact on energy consumption and costs.

Single Phase Immersion products are widely available. For example, Green Revolution Cooling (GRC), acknowledged as a leader in the field, has seen its technology deployed at several supercomputing sites, including facilities operated by the U.S. National Security Agency (NSA), the U.S. Air Force, and the Tokyo Institute of Technology. The company says its technology slashes datacenter cooling costs by as much as 95% while reducing overall power consumption by 50% or more. At a typical datacenter, switching to liquid immersion can also cut carbon output by 31%. "Immersion cooling reduces the cost, complexity, and the environmental impact of the world's digital infrastructure," says Gregg Primm, vice president of marketing for GRC.

In Two-Phase Immersion (TPI) cooling, electronic components are placed in a hermetically sealed enclosure filled with dielectric fluid. The electronics release heat into the fluid and cause it to boil at approximately 50 degrees Celsius.

c https://submer.com/submer-academy/library/dlc-direct-liquid-cooling/
d https://journal.uptimeinstitute.com/datacenter-pues-flat-since-2013/
e https://journal.uptimeinstitute.com/does-the-spread-of-direct-liquid-cooling-make-pue-less-relevant/

ACM Member News

THE INTELLIGENCE OF LARGE COLLECTIVES

"I fell in love with computer science during the first programming class I took; it was incredibly fun," recalls Radhika Nagpal, professor of Robotics at Princeton University in Princeton, NJ.

Nagpal earned her undergraduate, master's, and doctoral degrees in electrical engineering and computer science from the Massachusetts Institute of Technology in Cambridge, MA.

After obtaining her Ph.D., Nagpal joined the faculty at Harvard, where she remained until this year, when she joined the staff at Princeton.

"I'm interested in collective behavior overall, and how you coordinate the behavior of a large group of individuals to do something interesting and useful," Nagpal says, adding that those individuals could be things in biology like fish or ants, or they could be computers, distributed networks, robots in a warehouse, or self-driving cars.

Much of Nagpal's recent research focus has centered on self-managing robot swarms, especially in unstructured environments. Her lab has been working on an underwater robot swarm inspired by schools of fish, and she spent a recent sabbatical as an Amazon Scholar working with the retail giant's Robotics AI division on warehouse robots.

"If you look at Amazon warehouses, they have hundreds of robots operating together as one big collective, so robot swarms have become a reality," Nagpal explains.

Nagpal says when she started out, computer science was about computers, but it now reaches across multiple disciplines, as fields like engineering, robotics, and the social sciences have become more computational.

"That's one of the most exciting things happening in computer science today," Nagpal says.

—John Delaney

The resulting vapor condenses on a heat exchanger within the tank. The heat is transferred to water that flows outside the facility. The process, which offers the added benefit of using environmentally friendly non-flammable fluids, results in exponentially greater heat transfer than 1PI, though the approach remains in the early developmental stage.

Not surprisingly, the fluids themselves also are advancing. Although some dielectric substances used for immersion cooling are derived from mineral, vegetable, fluorocarbon, or synthetic oils, 3M and other companies have developed inert, fully fluorinated liquids that are clear, odorless, non-flammable, non-oil-based, low in toxicity, and non-corrosive. It is possible to match these products specifically to heat-transfer requirements. In addition, some of these dielectric fluids have been formulated for low global warming potential (GWP) and zero ozone depletion potential (ODP). Primm says GRC products utilize fluids that are biodegradable, non-toxic, designed to last 15 years or more, and are fully recyclable.

Eye on Immersion

If it sounds as though immersion cooling makes perfect sense—and significantly cuts costs in datacenters—a basic question arises: why hasn't the technology been widely adopted? Rogers says immersion cooling is viable, but it can be somewhat messy and involve ongoing operation and maintenance (O&M) expenses related to managing the systems and fluids. What's more, the upfront price tag for immersion cooling can be steep. Depending on the structure of an existing datacenter, the return on investment (ROI) is distant or non-existent, he points out.

Oak Ridge is among the organizations approaching immersion cooling methodically. It has adopted cold plate technology, but balked at adopting Single Phase Immersion cooling. "The biggest obstacle is substantiating the return on investment when, with existing technology, we can already capture over 95% of the heat from systems that are generating hundreds of kilowatts of waste heat." Rogers says there is no perfect approach: cold plates as well as enhanced water-cooling systems create their own sets of headaches, including introducing thousands of points of failure across large systems.

"Immersion cooling is ready to make a major impact. We're very close to reaching the tipping point where it will make a major difference."

Beran says that ultimately, most resistance is cultural; "There's a fear that systems will leak." However, immersion cooling has advanced to the point where the risk of a spill and contamination are remote. "While it's necessary to have a spill containment strategy in place, it really isn't all that different than having a fire extinguisher in your home. A lot of times people go their entire lives without using it, but it's there if you need it," he says. Other concerns, such as fluids dissolving stickers that display serial numbers, can easily be solved by laser etching numbers onto equipment, he notes.

Meanwhile, researchers continue to explore different technology components and frameworks for immersion cooling, as well as how to incorporate natural and synthetic fluids more effectively. For example, researchers are now looking for ways to enable the boiling of a cooling fluid directly in contact with electronic components.[f] There's also a focus on adapting and expanding the technology for battery systems, solar panels, and other devices that generate heat. For instance, one current research method centers on the use of modular jet oil cooling technology to draw heat from lithium-ion packs used in stationary electrical storage and transportation applications.[g]

To be sure, immersion cooling appears to be ready for prime time. In fact, vendors such as Submer now package systems in pods and micropods that are essentially plug-and-play, with coolant that can last 20 years.[h] Beran says the approach can cut operating costs by 33% compared to traditional air-based cooling.[i] Market research firm MarketStudyReport predicts adoption will grow by 24% from 2020 to 2025.[j]

"There are many ways to reduce energy consumption in datacenters," Beran concludes. "We're seeing new innovations and technologies emerge all the time. But immersion cooling is ready to make a major impact. We're very close to reaching the tipping point where it will make a major difference." ■

f https://www.sciencedirect.com/science/article/abs/pii/S0017931019336002
g https://asmedigitalcollection.asme.org/electrochemical/article-abstract/19/2/021007/1115735/Mineral-Oil-Immersion-Cooling-of-Lithium-Ion
h https://submer.com/immersion-cooling/
i https://www.delloro.com/is-immersion-cooling-the-answer-to-data-center-sustainability/
j https://www.marketwatch.com/press-release/immersion-cooling-market-share-key-growth-trends-major-players-and-forecast-2025-2021-10-12

Further Reading

Birbaraha, P., Gebrael, T., Foulkes, T., Stillwell, A., Moore, A., Pilawa-Podgurski, R., Miljkovic, N.
Water Immersion Cooling of High Power Density Electronics, ScienceDirect, Vol. 147, February 2020. https://www.sciencedirect.com/science/article/abs/pii/S0017931019336002

Trimbake, A., Pratap Singh, C., Krishnan, S.
Mineral Oil Immersion Cooling of Lithium-Ion Batteries: An Experimental Investigation, American Society of Mechanical Engineers (ASME) Digital Collection, May 2022, 19(2): 021007 https://asmedigitalcollection.asme.org/electrochemical/article-abstract/19/2/021007/1115735/Mineral-Oil-Immersion-Cooling-of-Lithium-Ion

Matsuoka, M., Matsuda, K., Kubo, H.
Liquid Immersion Cooling Technology with Natural Convection in Data Center, 2017 IEEE 6th International Conference on Cloud Networking (CloudNet), October 19, 2017. https://ieeexplore.ieee.org/abstract/document/8071539

Pérez, S., Arroba, P., Joya, J.M.
Energy-conscious optimization of Edge Computing through Deep Reinforcement Learning and two-phase immersion cooling, Future Generation Computer Systems, ScienceDirect, July 31, 2021. https://www.sciencedirect.com/science/article/pii/S0167739X21002934

Samuel Greengard is an author and journalist based in West Linn, OR, USA.

DOI:10.1145/3532632

Peter J. Denning

The Profession of IT

Involvement and Detachment

How detachment from your community blocks your success at leading innovations, and what to do about it.

OUR AGE VALUES abstraction, the characterization of large populations with statistics, properties, and rights. This is natural for governments, which are preoccupied with defining and dispensing services efficiently across large populations. Global connectivity enables data collection from everyone and distillation of trends in large groups, revealing large-scale phenomena that were not visible in prior times. For example, COVID is treated as a large-scale phenomenon of infection, hospitalization, and herd immunity through vaccination. In contrast, when the Spanish Flu epidemic began in 1918, there were no centers for disease control, no health oversight agencies, no daily communications about the spread of the disease, no ability to exercise large-scale controls. The ability to view large-scale phenomena through the lens of distilled data is strong force for abstraction. Unfortunately, abstraction is also a force for detachment, the loss of connection with fellow human beings.

In my work, I coach graduate students on their innovation projects. Many get stuck, unable to get their communities to engage with them. An invisible force seems to thwart them from achieving their innovation goals. This was puzzling because they seemed to be doing the right things: looking for concerns, crafting good envisioning stories, and making offers. Then I discovered a distinction that revealed the invisible force. It is the distinction between the moods of involvement and detachment.

I call them moods because they are hidden dispositions that orient how we engage with our communities. Because we do not notice them, we cannot see how detachment stifles engagement and involvement empowers it. By becoming aware of these moods, we can put the distinction to work in our own professional life.

Detachment orients us to be an outside observer of our community. When we undertake to find an innovation that resolves a community issue, it is easy to take the stance of an outside expert who can see the problem that the community cannot see because of their immersion in it. A big problem with this orientation is that we substitute our belief about what they need for whatever concerns they are experiencing. In our certainty that our solution will work we become impatient with their uncertainty about whether anything will work. Our outsider solution looks to them like an algorithm rather than a compassionate understanding of their situation. They do not trust us and pull away from engaging with us.

Involvement, the opposite of detachment, orients us to a deep listening to our community's concerns, even when community members are unable to put them clearly into words. We are curious about their practices, their histories, their daily activities. We are concerned about their well-being. We offer ourselves as a compatriot in their issues, aiming to serve them with a new practice that will dispel their problem. We are not an outsider, we are a fellow member of their community. They come to trust us and want to engage with us.

It is worth discussing the roots of detachment so that we can see why we

IMAGERY BY MASTER1305; COMPOSITION BY ANDRIJ BORYS ASSOCIATES

are drawn into it more than we would like. Then we can see what basic practices will support our involvement.

Objectivity versus Detachment

We often say that some professions, such as science or the law, require objectivity, an ability to look dispassionately at a phenomenon and the evidence for and against it. This kind of objectivity is an ability to manage your prejudices and biases; it is often seen as a good thing even if it is difficult to attain. But there is another side to objectivity. We view people and their actions as objects—resources that can be efficiently controlled by a set of prescribed rules. Let's call these "anti-bias objectivity" and "control objectivity." Control objectivity can support detachment and anti-bias objectivity can support involvement.

Bureaucracies are vaunted examples of detached organizations that excel at control objectivity. Bureaucracies are needed to provide services that governments have promised to their people. They treat everyone the same and grant no waivers or exceptions to the rules. They are designed to be efficient automation machines, dispassionately dispensing services. Many people are dissatisfied with bureaucracies, which have no compassion for individual circumstances, make frequent mistakes, and provide no customer service to correct mistakes or enact reforms. Bureaucracies are highly detached practitioners of control objectivity.

The Draw of Detachment

Our age is also one of great reverence for science. The ideal of science is the unbiased and unemotional observer seeking to objectively understand what is going on and predict what will go on in the future. This form of observer is "outside" the phenomenon, looking in. Even though detachment is also outside, detachment includes a sense of knowing the answer whereas objectivity is looking for an answer. In science, objectivity is important for the scientific method, the ability to be a standard observer who finds explanatory patterns that can be reproduced by others. Data is a powerful tool for objective abstraction.

Hubert Dreyfus, a philosopher, called this sort of detached objectivity a "technological way of being" because it sees the whole world as objects that can be investigated, manipulated, predicted, and controlled by technology. This worldview can lead to chronic feelings of disorientation because many things in the world cannot be manipulated, predicted, or controlled.

Involvement makes no distinction between "inside" and "outside." We participate fully in the practices, concerns, norms, and values of a community. We grow and nurture relationships with those in your community. We make commitments to take care of concerns in our community. We take responsibility for our commitments and the consequences of our actions. There is no way to separate our "observer" from our community: much of what we think is "inside" us is actually the manifestation of our community in our bodies. We are not the outside observer looking in; we are an integral, functioning part of our community.

The philosophy of detached objectivity traces back to the philosopher René Descartes. He lived in the time of the Thirty Years War (1618–1650), in which he and many others longed for a way out of war. He thought the way out would be to avoid getting swept up into emotional confrontations and instead resolve problems though rational discourse. The cornerstone of his philosophy is mind-body dualism—the mind is rational, the body's emotions and desires are not. Descartes held that the body is constantly dragging the mind away from rational thought through emotions and base desires. Fifty years later, the German polymath Gottfried Leibniz, known to us as the inventor of calculus, sought an algebra of discourse that would allow people to reach rational conclusions by "calculating together." Although Descartes and Leibniz disagreed on many things, they both believed that mathematicians and others trained in logical, rational thought are better positioned to find rational resolutions of otherwise emotional confrontations. This philosophy became very popular and is still respected and revered today. Mostly, we are oblivious to the beliefs of this philosophy gives us and simply accept them as truths about the world.

Two Faces of Science and Technology

Though we are told that detachment is an ideal of science, a closer look reveals that scientists live in a highly involved world. In his book *Science in Action* (Harvard 1987), the philosopher and sociologist Bruno Latour distinguished between "ready-made science" and "science in the making." Ready-made science is the settled laws and theories of science that anyone can use and fully trust without having to understand all the details behind them. For example, Einstein's formula $E=mc^2$ came from Relativity Theory and has been extensively validated in experiments by highly trained physicists. Anyone can use the formula at any time and trust it fully without knowing the math or experiments behind it.

In contrast, science-in-the-making is a messy and chaotic process of hypothesis building, testing, controversies, and recruiting allies. The scientific literature records many heated battles between scientists as they struggled to understand what the truth is. Over time, a controversial hypothesis can evolve into a trusted law as more and

> **Involvement makes no distinction between "inside" and "outside."**

more experiments validate the claim, and the doubters fall away.

Latour depicts this dual nature of science with an image of the two-faced Roman God, Janus. One face, seasoned and creased with lines of wisdom, looks back over all that has happened and tells us what is true and repeatable. The other face, youthful and brash, looks forward and tries to make sense of the unknown ahead. These opposing faces embody inverted interpretations of the world. To illustrate the differences, Latour contrasts ready-made with in-the-making with aphorisms such as "When things are true, they hold" versus "When things hold, they start becoming true;" and "Science is stronger than the multitude of opinions" versus "Decide which opinions are worthy of consideration." The detached, ready-made views are all statements of certainty, whereas the involved, in-the-making view are about uncertainty. Detachment seems to be a poor stance for making headway with the uncertainties that scientists are called to resolve. The most successful scientists embrace their immersive involvement in science-in-the-making in order to achieve the detachment of ready-made science. This conclusion is not limited to science: detachment seems to be a poor stance for making headway with uncertainties and yet characterizes the solutions that emerge.

In this way, Latour artfully answers the question: How can science be detached and involved, methodical and chaotic, at the same time? Even though they co-exist, the two views are in constant tension, the one pulling against the other. Both are necessary, but not easy to navigate.

Do you see a resemblance between this account of science and your work of solving problems with your clients? To make headway, you must be fully involved in the community and the concerns you are taking care of. You constantly face uncertainty, doubt, and resistance. When your work is finished, you enjoy pride in a job well done. You can sit back, detached, and say the work was good. Detachment is good for reflection, involvement navigating the uncertainty. You move back and forth between involvement and detachment.

Resolving the Dilemma

Detachment and involvement are both useful stances, depending on the situation. A detached mood is appropriate for scientific investigations, law enforcement, bureaucracy management, and jury trials. An involved mood is appropriate for making headway in science and engineering, design, community work, and leadership. Yet our history of detachment exerts a strong pull, often so strong that we cannot become involved even though we want to. How can we open ourselves to greater involvement?

The kernel is service and care. Ron Kaufman, known worldwide for his teachings on uplifting service, has proposed a useful insight. He defines service as action that brings value to someone. He defines care as a concern for someone's future well-being. These two ideas blend together when the value of service is the well-being of others. He summarizes the blend as "service is care in action." This applies directly to our professional work. We are at our best when we make our expertise available to take care of the well-being of others.

Thus, in answer to the question at the start of this column, it is right to care about our communities and be drawn into involvement with them. Then our offers organize our actions to be of service to our communities.

We hold service and care as the ideal of innovation leadership. Some innovations do not meet this ideal because care is missing. An oft-cited example is the practice of Internet companies of selling personal data of their customers to maximize revenue.

A core practice for service and care is empathetic listening. This is an ability to listen for deep, unarticulated concerns of people we have conversations with. Can we ferret out their concerns? Do we have a curiosity about them and what skills and talents they bring? Do we seek to learn their history? Their interests? What provokes them? What they care most about? How the world looks from their perspective? When we can give voice to what they care about, our offers are attractive.

Beware that our background breeding in detachment can sidetrack us in listening empathetically. One way this can happen is to treat listening as a technique rather than a sensibility and skill. For example, "active listening" is a technique where we repeat back what we thought the other person said so they can validate that we "got it." Unfortunately, this puts us in the mindset of a tape recorder and distracts us from listening to concerns concealed behind the words spoken. A second way detachment can sidetrack us is its familiarity. When we are in the habit of being detached, talking to people about their concerns may seem like heavy work. To avoid the work, we turn to intuition and logic to deduce what they are concerned about. We imagine what the other person ought to care about, substituting our "concern for them" for their actual concern. It often turns out that the concerns we imagine are not the ones they really care about—and no wonder our offers fall on deaf ears.

Thus, detachment can draw us into a condition that might be called "self-service is self-care in action," which is not what involvement is about.

The bottom line is this. To open yourself to true involvement, engage in many conversations with your community, listening behind their words for what they care about. Give voice to their concerns. Make offers that improve their well-being relative to their concerns. Fulfill your offers. Do not let your imagination, honed by years of detachment, substitute your ideas for their concerns. Be involved!

Peter J. Denning (pjd@nps.edu) is Distinguished Professor of Computer Science and Director of the Cebrowski Institute for information innovation at the Naval Postgraduate School in Monterey, CA, is Editor of ACM Ubiquity, and is a past president of ACM. The author's views expressed here are not necessarily those of his employer or the U.S. federal government.

I thank Dorothy Denning, Elisa Caeli, Ron Kaufman, John King, and Todd Lyons for conversations and insights on this topic.

DOI:10.1145/3532631

Peter G. Neumann

Inside Risks

Toward Total-System Trustworthiness

Considering how to achieve the long-term goal to systemically reduce risks.

COMMUNICATIONS' **INSIDE RISKS** columns have long stressed the importance of total-system awareness of riskful situations, some of which may be very difficult to identify in advance. Specifically, the desired properties of the total system should be specified as requirements. Those desired properties are called emergent properties, because they often cannot be derived solely from lower-layer component properties, and appear only with respect to the total system. Unfortunately, additional behavior of the total system may arise—which either defeats the ability to satisfy the desired properties, or demonstrates that the set of required properties was improperly specified.

In this column, I consider some cases in which total-system analysis is of vital importance, but generally very difficult to achieve with adequate assurance. Relevant failures may result from one event or even a combination of problems in hardware, software, networks, operational environments, and of course actions by administrators, users, and misusers. All of the interactions among these entities need to be considered, evaluated, and if potentially deleterious, controlled by whatever means available. The problem to be confronted here is trying to analyze an entire system as a composition of its components, rather than just considering its components individually. In many cases, system failures tend to arise within the interactions and interdependencies among these components, depending on whether a system was designed modularly to minimize disruptive dependencies, with each module carefully specified.

Addressing this problem is a daunting endeavor, even for seasoned developers of critical systems. Whether explicitly defined or implicit, total-system requirements may be highly interdisciplinary, including stringent life-critical requirements for human safety; system survivability with reliability, robustness, resilience, recovery, and fault tolerance; many aspects of security and integrity; guaranteed real-time performance; forensics-worthy accountability, high-integrity evidence, and sound real-time and retrospective analysis; defenses against a wide ranges of physical, electronic, and other adversities; and coverage of numerous potential risks. Ideally, requirements should be very carefully specified at various architectural layers, preferably formally as much as possible in newly developed systems, and especially in particularly vulnerable components. Although this is usually not applicable to legacy systems, it is stated here as a farsighted goal for future developments.

Total-system architectures that must satisfy high-assurance requirements for trustworthiness may necessarily encompass much of what is described here. However, when executed on untrustworthy hardware and untrustworthy networks, the behavior of operating systems and application software should be considered with suspicion, as it suggests desirable emergent properties of the total system may have been compromised, or could easily be (resulting in adverse behavior).

An almost self-evident conclusion is that total-system trustworthiness with respect to realistic requirements under realistic assumptions is a very long-term goal that can never be completely achieved with any realistic sense of assurance. However, many efforts in that direction would be extremely valuable in attempting to withstand many adversities that are uncovered today. Several efforts currently under way are noted in this column, and seem to be small steps in that direction for new systems, although as previously mentioned, much less applicable to existing legacy systems. However, the enormity of the entire challenge should not discourage us from making structural improvements that could help overcome today's shortsightedness.

Hierarchical Layering and Formal Methods

Hierarchically layered designs have considerable potential, but today are found mostly in well-designed operating systems and layered networking protocols. The concept has often been rejected because of erroneous nested efficiency arguments that can be overcome through good design practice. Formal specification languages and formal analysis of software and hardware have become much more widely applied in recent years. Formal specifications can also exist for system requirements, high-level system architectures, hardware ISAs, and actual hardware. Here are just a few early examples (many others are omitted for brevity).

- Dijkstra's THE system[4] provided a conceptual proof that a carefully layered hierarchical locking strategy could never cause a deadlock between layers—although a deadlock within a single layer was discovered years later).

Addressing this problem is a daunting endeavor, even for seasoned developers of critical systems.

- David Parnas's seminal work on encapsulated abstraction presented advanced design considerations in the early 1970s. It has become a vital part of structured developments.[11]
- The SRI Hierarchical Development Methodology,[13] which was the basis for PSOS[9] (in which seven layers of hardware abstractions and nine layers of system software were formally specified in a non-executable language, so that proofs could have been sewn together for each layer based on their lower ones. PSOS made extensive use of Parnas's work. (Many other methodologies are also used more widely, but mostly with less rigor.)
- Virgil Gligor[5] spearheaded some contemporaneous efforts to formalize higher-layer policy issues in 1998. The Clark-Wilson paper[2] extended the notion of security requirements into an informal representation of generic application-integrity principles.
- More recently, seL4[7,8] and CertiKOS[6] provide significant advances in software hypervisors (the latter internally layered approximately similar to HDM).
- The new CHERI-Arm Morello hardware instruction-set architecture with multiple operating systems[14] includes proofs[10] the ISA satisfies several critical hardware trustworthiness properties. Hardware Morello chips and system-on-chip boards are currently being made available for experimental use by Arm Ltd. That effort is only one step toward trustworthy hardware; CHERI-RISC-V is also specified. As with all other hardware, the consistency of the actual Morello hardware with its ISA specification remains unresolved—that is, the hardware must do exactly what is specified and nothing else (for example, no supply-chain

compromises such as added Trojan horses resulting from inserted analog circuitry[15]). In addition, Nirav Dave's Ph.D. thesis[3] extended the Bluespec executable specification language (BluespecSystem Verilog BSV) to BSL, which could enable an extended compilation into both hardware (the lower layers) and software (the upper layers).

A long-term goal for the future would be to have hierarchical proofs (from the hardware up through hypervisors, operating systems, and application code) to prove that specified total-system requirements (with stated assumptions) could be satisfied with some desired measure of assurance. There are still many potential pitfalls (incomplete requirements and specifications, inadequate development and assurance tools, sloppy programming, unreliable systems, malicious attacks, and so forth). However, assurance is necessary for each step along the way to this goal, as well as better analyses of entire systems. Unfortunately, that approach is not applicable to most legacy hardware-software systems, which suggests the long-term approach must be injected early into future technology developments.

Perhaps as an indication that more R&D is needed, DARPA is currently planning a new program called PROVERS: Pipelined Reasoning of Verifiers Enabling Robust Systems for extending formal methods to work at the scale of real systems.

Illustrative Applications

Several relevant application areas in which the compromise of total-system attributes may be of great concern are considered here. In each case, there are difficulties in analyzing the relevant components, but also their embeddings into total systems; proving anything convincingly could be very difficult—if not generally impossible. Furthermore, even if a particular component could somehow be shown to be logically sound by itself (which often seems not to be the case), its compliant total-system behavior may be compromised by exploitation of hardware and operating system flaws that can undermine its integrity, or by poor application programs.

- *Cryptography*. Cryptography is sometimes thought of as a panacea. However, overreliance on the very best cryptographic implementations and their applications is especially worrisome, particularly when embedded in hardware or operating systems that can themselves be compromised.
- *Real-time systems*. The design of real-time systems with guaranteed performance and fail-safe/fail-soft/fail-secure requirements must anticipate a much wider range of faults and failure modes than laptops. The same is true of some analog-digital and mixed-signal cyberphysical systems. Again, low-level failures can compromise the ability to satisfy those requirements, as can simple application-specific code.
- *Election integrity*. Previous Inside Risks columns on election integrity have stressed that every part of the election process is a potential weak link. Existing commercial systems are seriously flawed, and many of the overall systemic weaknesses are external to computer systems and can make the technology more or less irrelevant if the results have been compromised.
- *Quantum computing*. In the future, quantum computing and its integration into networking with conventional computing are likely to be fraught with unanticipated problems. Also, the necessary error-correcting coding required in quantum computers may miscorrect results whenever the errors exceed the limits of the coding system. Thus, the choice of the coding system to fit the actual range of hardware failures becomes critical.

The enormity of the entire challenge should not discourage us from making structural improvements that could help overcome today's shortsightedness.

▸ *Multilevel security*. One of the most demanding areas of trustworthy computing and communications involves being able to concurrently deal with different levels of critical security (for example, top secret to unclassified). With very few exceptions, most of the efforts in the 1970s and 1980s assumed implementing the required separation in a software kernel would be good enough. Unfortunately, the available hardware was (and still is) inadequate. Notable exceptions to the software-only approach included Butler Lampson's BCC-500 computer (Berkeley Computing Corp.) in the late 1960s, the hardware-software MLS retrofit for Multics in the early 1970s, and PSOS (which sought to ensure MLS as a strongly typed hardware capability extension) in the mid-1970s.

▸ *Artificial intelligence*. The trustworthiness of systems based on deep learning, neural networks, and many other aspects of what is generally referred to as artificial intelligence is typically difficult to prove or otherwise evaluate, for all possible circumstances. Also, AI elements that require self-adaptation or training may have not been programmed or trained properly for their intended use; also, algorithms and training data may be intentionally or inadvertently biased. Certainly, a trained neural network can do no better than the data it is fed. In generally, the use of AI would seem very risky in life-critical and other systems with stringent requirements—especially where deterministic or demonstrably sound results would be essential (see for example, Parnas[12,a]). Nevertheless, AI is very popular, and is finding many diverse useful applications.

These examples expose just tips of multiple icebergs, but are intended to be suggestive of the difficulties that must be overcome.

In each of these cases, there is also a desirability of having some independent sanity checks ensure the total-system results are correct—or at least within realistic bounds—with respect to the stated requirements. An analogy in formal theorem proving is to use trustworthy proof checkers to check the proofs, although that still assumes the underlying assumptions and the proof checkers are correct and unbiased.

a This article refers to many additional references that deserve to be included here, such as Parnas's remarkably prescient early papers.

There is also a desirability of having some independent sanity checks ensure the total-system results are correct.

What Is Needed?

The desiderata were established many years ago, but are still not used widely in practice. A grossly oversimplified set might include something like this:

▸ Consider established principles for total-system development and trustworthiness, and invoke those that are most relevant.

▸ Establish well-defined total-system requirements against which evaluations can be made, and specify them formally where possible.

▸ Establish well-defined system architectures, hierarchically defining accessible interfaces at each major interface, from hardware to operating systems and applications (for example, Robinson-Levitt,[13] which was applied to conceptual hardware and software in PSOS, and to the Ford Aerospace KSOS1 MLS kernel in software.

▸ Use formal specifications and formal methods by which formal analysis is possible, particularly in systems with particularly critical requirements. Analysis might include dependency analysis (seeking dependence only on less-trustworthy entities, and avoiding circular dependences that might cause deadlocks), and proofs of essential properties. Hierarchical proofs from the ground up are theoretically supported,[13] but still lurking in the future if they were to span hardware, operating systems, applications, and total-system requirements as much as possible—leaving out-of-scope assumptions clearly stated via itemization of unaddressed or missing requirements, and enumerating threats that remain uncovered.

▸ Address myriad other problems proactively throughout.

Conclusion

Significant progress is being made with some of the steps toward the desired long-term goal of total-system trustworthiness. Of course, all of this is still nowhere near enough, considering all the extrinsic problems we face. However, the goal is nevertheless worth pursuing for new critical systems—to the extent it is realistic. This suggests we must begin now to recognize the relevance of the overall long-term goal. C

References

1. Berson, T.A. and Barksdale, G.L., Jr. KSOS: Development Methodology for a Secure Operating System, National Computer Conference, *AFIPS Conference Proceedings 48* (1979), 365–371.
2. Clark, D. and Wilson, D.R. A comparison of commercial and military computer security policies. In *Proceedings of the 1987 Symposium on Security and Privacy*. IEEE Computer Society, Oakland, CA (Apr. 1987), 184–194.
3. Dave, N. A Unified Model for Hardware/Software Co-design, MIT Ph.D. thesis. 2011.
4. Dijkstra, E.W. The structure of the THE multiprogramming system. *Commun. ACM 11*, 5 (May 1968), 341–346.
5. Gligor, V.D. and Gavrila, S.I. Application-oriented security policies and their composition. In *Proceedings of the 1998 Workshop on Security Paradigms*, Cambridge, England, 1998.
6. Gu, R. et al. CertiKOS: An extensible architecture for building certified concurrent OS kernels. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)*. (Nov. 2016), 653–669.
7. Heiser, G., Klein, G., and Andronick, J. seL4 in Australia: From research to real-world trustworthy systems. *Commun. ACM 63*, 4 (Apr. 2020), 72–75.
8. Klein, G. et al. Comprehensive formal verification of an OS microkernel. *ACM Transactions on Computer Systems 32*, 1 (Feb. 2014).
9. Neumann, P.G. et al. A Provably Secure Operating System: The System, Its Applications, and Proofs. SRI International. 1980.
10. Neumann, P.G. Fundamental trustworthiness principles in CHERI. In A. Shrobe, D. Shrier, and A. Pentland, Eds. *New Solutions for Cybersecurity*. MIT Press/Connection Science (Jan. 2018); https://bit.ly/3kgxyt6
11. Nienhuis, K. et al. Rigorous engineering for hardware security: Formal modeling and proof in the CHERI design and implementation process. In *Proceedings of the 36th IEEE Symposium on Security and Privacy*, May 2020.
12. Parnas, D.L., Clements, P.C., and Weiss, D.M. The modular structure of complex systems. *IEEE Transactions on Software Engineering SE-11*, 3 (Mar. 1985), 259–266.
13. Parnas, D.L. The real risks of artificial intelligence. *Commun. ACM* (Oct. 2017).
14. Robinson, L. and Levitt, K.N. Proof techniques for hierarchically structured programs. *Commun. ACM 20*, 4 (Apr. 1977), 271–283.
15. Watson, R.N.M. et al. Cambridge-SRI CHERI-ARM Morello and CHERI-RISC-V; https://www.cl.cam.ac.uk/research/security/ctsrd/cheri/
16. Yang et al. A2: Analog malicious hardware. In *Proceedings of the 2016 IEEE Symposium on Security and Privacy*. IEEE Computer Society.

Peter G. Neumann (neumann@csl.sri.com) is Chief Scientist of the SRI International Computer Science Lab, and has moderated the ACM Risks Forum since its beginning in 1985.

DOI:10.1145/3532633

George Neville-Neil

Kode Vicious
The Planning and Care of Data

Rearranging buckets for no good reason.

IMAGE FROM SHUTTERSTOCK.COM

Dear KV,

After several years as a startup, we seem finally to have gotten large and popular enough that management and legal are looking at how we store and segregate our user data. To say this feels like a fire drill would be understating it. Everyone now seems to have an opinion on how we should handle user data. Meetings on this topic would be funny if they were not so tragic. It is not as if we are a huge company that claims billions of users, and, of course, it is important to protect our customers' data. But all the hand-wringing at this point seems to be very late in the game and is likely to end up causing a huge amount of engineering that ultimately does not add value to the product but is, instead, a way for management—or perhaps the legal department—to protect themselves. I cannot imagine this is of value, but maybe you have a different opinion? I suspect at the very least, you do have an opinion that will be more fun to read than the email messages from the legal department. I feel as if we are just rearranging buckets for no good reason.

Bucketed for No Good Reason

Dear Bucketed,

In a world that now contains so many rules and regulations around how a company handles user data, I would like to say I am surprised your company managed to go several years before reaching this juncture. But bad news rarely surprises me, even less so than stories of people not thinking about how to handle the data they receive.

It is not just rules and regulations that ought to cause people to think about data engineering and data maintenance; it is the fact we have now come to a place in computing where data has significant value and significant risk—in equal measure. A wobble down memory lane shows us the trajectories of engineering efforts through computing have changed significantly over the past 70 years. While 70 years might be considered a short time in some of the traditional sciences, the amount of change over that time in what matters to people working with computers has been dramatic. We have moved from the 1950s and 1960s, where hardware was the dominating cost and the focus of our efforts, to the rise of software in the latter part of the 20th century, to the rise of data in the early 21st century. Why?

Moore's Law has a lot to answer for here, as well as the human inability to throw stuff away once we have collected it. Parkinson's Law ("Work expands so as to fill the time available for its completion") has a corollary, "Data expands to fill the space available for storage," which has been the case ever since we have had the ability to store data.

I remember when I was younger visiting my uncle's office at his university, where he had stacks and stacks of punch cards.

"What are these?" I asked.

"That's all my astrophysical data for my research," he explained.

My uncle had only limited space for

his boxes of punch cards, so I encouraged him, as a know-it-all 16-year-old KV, to switch to tape. I never asked if he did, but I bet if he did, he would have wound up with even more data than would fit in the cubic feet available in his office.

As time progressed, software came to dominate the cost of systems because computers became less expensive and more powerful, and, therefore, we could write larger and more-complex programs, which then became systems of programs, and then distributed systems of programs.

All this increasing complexity forced us to find solutions to a software crisis that was well-described by Dijkstra in his 1972 ACM Turing Lecture: "But instead of finding ourselves in the state of eternal bliss of all programming problems solved, we found ourselves up to our necks in the software crisis! How come? The major cause is that the machines have become several orders of magnitude more powerful! To put it quite bluntly: as long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become an equally gigantic problem." —Edsger Dijkstra, *The Humble Programmer*; https://bit.ly/3JCvc2P

The software crisis has never abated, no matter what. Often, ridiculous trends have appeared to supposedly address it. Modular programming, object oriented programming, pair programming, Agile, Scrum, and other approaches, were all meant to address the fact the hardware—and particularly the software—we were building was, and continues to be, too complicated for those of us who work with it to understand.

As compute and memory got cheaper, so did storage. In the 1980s, the early micros could store a few hundred kilobytes of data on floppies, or, if we were rich, we might have a 10MB drive in a PC. Twenty years later, which is now 20 years ago, that went to many gigabytes of storage, and now it is terabytes—and that is what we can *personally* store. Datacenters, of course, went through similar, spectacularly quick growth in storage space.

It is not just the amount of data we are storing; it is the relationships among the data. The relationships drive the complexity, just as the explosion of libraries and packages used in modern software drives up the complexity and cost of software systems.

The software crisis has never abated, no matter what.

What does all this mean in 2022?

It is well past the time when everyone who even thinks about collecting data, user or otherwise, must first think seriously about data engineering and data maintenance, because the costs of getting it wrong are far too high—both monetarily and societally. I would like to say no sane person simply sits down and starts typing code—with just a loose idea in mind—and expects things just to work out in the future. What goes for software engineering goes for data engineering. You really cannot just dump data into a cloud bucket or any other large storage system and expect everything will work out for the best.

There are people who have thought about this for a long time, but they often do not have much sway anymore. Before the rise of inexpensive compute and all the nontransactional database systems, we had people who were specialists in how data should be stored, and these people were necessary for an efficient, data-storage back end. These were the database administrators, but these people are rarely involved in getting startups going because the startups see code first and data second, unless their real go-to market is to get one of the FAANG—Facebook (now Meta), Amazon, Apple, Netflix, Google (now Alphabet)—to buy them for the value of that data. Even then, they are more like vacuum cleaners, sucking up everything they can get a hold of, with little concern for its safety, future value, and risk.

Even when companies start down the right path, they usually fail at data maintenance, just as companies fail at software maintenance. New data is accreted without plans and it piles up everywhere because people figure they will just sprinkle on some machine-learning magic and get more value out of it.

There are no magic bullets in engineering. If you slap an extension on a house without thinking about its effect on the overall structure, your extension, or the entire house, is going to be damaged and, in the worst case, come crashing to the ground. Our industry is littered with these data corpses, but a little bit of planning at the start and care throughout the lifetime of the data will pay off handsomely.

Questions such as, "How do we secure this data?" work only if you ask them at the start, and not when a group of lawyers or government officials are sitting in a conference room, rooting through your data and logs, and making threatening noises under their breath. All the things we care about with our data—security, privacy, efficiency of access, proper sources of truth—require forethought, but it seems in our rush to create *stakeholder value* (a term often used to justify so much) we are willing to sacrifice these important attributes and just act like data gourmands.

Now that data has surpassed most software in size and complexity, it is time to make data engineering and data maintenance first-class topics of study. To do anything else simply invites us to make the same mistakes and put people and our companies at risk.

KV

Related articles on queue.acm.org

The Case Against Data Lock-In
Brian W. Fitzpatrick and JJ Lueck, The Data Liberation Front
https://queue.acm.org/detail.cfm?id=1868432

IoT: The Internet of Terror
Kode Vicious
https://queue.acm.org/detail.cfm?id=3121440

Federated Learning and Privacy
Kallista Bonawitz, Peter Kairouz, Brendan McMahan, and Daniel Ramage
https://queue.acm.org/detail.cfm?id=3501293

George V. Neville-Neil (kv@acm.org) is the proprietor of Neville-Neil Consulting and co-chair of the *ACM Queue* editorial board. He works on networking and operating systems code for fun and profit, teaches courses on various programming-related subjects, and encourages your comments, quips, and code snips pertaining to his *Communications* column.

DOI:10.1145/3503916
Bran Knowles, Kelly Widdicks, Gordon Blair, Mike Berners-Lee, and Adrian Friday

Viewpoint
Our House Is On Fire

The climate emergency and computing's responsibility.


IMAGE BY MASTER1305

We are writing this as the world's leaders gather at the UN Climate Change Conference (COP26). In today's news, Boris Johnson is "upbeat," reporting that if this were a football match, the world is down only 5-2 or 5-3, as opposed to 5-1 only a few days earlier. As China's leaders (conspicuously absent) haggle over whether the target should be 2 degrees Celsius warming instead of 1.5 degrees, and nations engage in a pledge drive to reach an unfathomable 28 gigatons emissions reductions by 2030, it is easy to lose sight of what is really at stake here. We are talking about the risk of catastrophic climate change and whether we going to have a planet habitable for human life.

What is computing's pledge? Beyond being keen to innovate digital 'solutions,' are we going to address our contribution to the climate emergency? ACM recently released its first TechBrief,[1] which was designed to communicate to an audience of policymakers some of the key headlines regarding the climate impacts of computing. The brief was in part a response to proposed climate strategies that entail investment in digitalization based on unproven climate gains without acknowledgment of the carbon costs of such endeavors. The overall message of the piece is that computing is by no means immaterial,[10] and given that computing's emissions are rising, we cannot assume that continuing to do what we have been doing is going to produce a sudden reduction in computing's footprint.

This Viewpoint draws in part from a much longer report[7] that elaborates at length the kinds of details that matter in this space. Estimates of computing's current and future carbon footprint vary, and there is (sometimes heated) disagreement about which figure to accept as 'fact.' For example, the percentage of global energy use by datacenters ranges from approximately 1%–3%. But these estimates are exactly that: estimates. Anyone claiming to know precisely the carbon footprint of something as vast and as multiplex as the world's datacenters, or networks, or devices should be met with skepticism. There is simply too much interpretative license involved in setting the boundaries of the analysis, and too little formal and transparent accounting. We can quibble whether computing's global share of carbon emissions is closer to 1.8% or 2.8%—or possibly even higher (around 3.9%) if accounting for the full supply chain and complete life cycle of the technologies[7]—but in doing so we are avoiding reckoning with the hard truths about computing's responsibility.

We will most likely never do better than a best guess at computing's carbon footprint, but given uncertainties it would be safer and more responsible to act on the assumption that higher estimates could be closer to the truth—

especially since the pace of warming has exceeded our expectations at every point. But in big-picture terms, the difference between 1.8% and 3.9% does not fundamentally change our mission: computing's emissions must be reduced urgently and drastically. How are we going to achieve this?

Through Efficiency?

There is a natural logic to reducing emissions by using less energy for any given operation, and it would be highly convenient for computing to be able to claim environmental benefits for continuing to deliver efficiency improvements (our bread and butter). Unfortunately, however counterintuitive, the arc of computing does not bend toward lower emissions as underlying efficiency improves; quite the opposite. Computing's footprint has risen steadily despite becoming much more efficient in the transmission and storage of data over the last 50 years. Greater efficiency leads, almost always, to growth in overall carbon emissions, as efficiency gains are quickly swamped by the desire to do more; hence we see global emissions increasing decade after decade despite continual efficiency gains in every sector.

In spite of these issues, computing technologies are considered critical to enabling key 'emissions-reducing' efficiency improvements across the economy, and as a result, the entire computing sector basks in the green glow reflected off these ostensibly honorable pursuits.[4] The fact is, the vast majority of computing solutions are *additive* in terms of carbon. Of course some digital innovations specifically intend to increase efficiency of some process and/or reduce emissions, but most innovations are introduced in the service of generating profit and can claim no environmental benefit. Responsible individuals are 'doing their bit' to reduce their personal footprints by cycling to work, using electricity during off-peak hours, going vegan, and so forth; meanwhile, computing is "repeatedly finding ways to use more chips in parallel,"[5] without a care for environmental costs, and the footprints of cryptocurrencies soar past that of ever-larger nations. This exposes the lie behind "micro-consumerist bollocks,"[8] and illustrates clearly why computing cannot be allowed to shirk its responsibility. In the absence of some external constraint (such as a carbon price applied at the point of extraction), the efficiencies computing delivers are highly unlikely to materially reduce emissions, particularly not at the scale required. However, efficiencies delivered by computing technology could play a vital role in enabling continued functionality within a resource-constrained future. The computing industry should be lobbying strongly for the introduction of a carbon constraint! Far from stymieing innovation, it requires it, and guarantees the efficiencies computing can deliver will be more valuable than ever.

[A carbon constraint] guarantees the efficiencies computing can deliver will be more valuable than ever.

Through Renewables?

One of the many frustrating things about this crisis is that we have known for decades what we need to do to solve it: we need to leave fossil fuels in the ground and ensure the immediate development and deployment of clean energy. From a technical perspective, it looks challenging but possible to replace today's energy supply. Yet this will not happen overnight, and there are significant embodied carbon costs in manufacturing these technologies, as well as other devastating environmental and humanitarian impacts of extracting the necessary (and dwindling) raw materials.

Thus, a renewable energy infrastructure gives us some respite to meet climate targets while continuing to use energy, but it does not grant us freedom to meet ever-increasing energy demand. Using this limited renewable energy supply to fuel computing's unchecked growth reduces other sectors' ability to decarbonize.[7] A serious and proportional response to the climate emergency would, therefore, involve constraining energy demand and mitigating drivers of infrastructure growth, and as a result, also consuming less energy. In real terms for computing, this means manufacturing fewer devices, storing and processing less data, generally managing with less compute power; and in terms of technical ambitions, scaling back the Internet of Things, resisting the temptation to throw AI and blockchain at every problem, and breaking free of the cycle of ever increasing demand for computation.[3]

Through Offsets?

Major tech companies like to boast of being "carbon neutral"—Google, for example, claiming to have eliminated in 2007 its entire carbon legacy. This is a nice soundbite, designed to mislead the public into thinking Google has achieved zero emissions every year since 2007. To the contrary, they have 'offset' the approximately 20 million tons they have emitted since then by paying toward capturing escaping natural gas.[6] Critiques of the true impact of offsets aside, when the goal is to reverse an as-yet undented curve of relentless exponential growth in carbon emissions, we need to be planting trees, capturing natural gas, and employing all possible techniques to sequester carbon *while drastically reducing emissions*. Offsetting is better than nothing, but not nearly as good as not emitting in the first place.

We must also call out other forms of "buying indulgences out of environmental guilt."[5] When Amazon's Jeff Bezos pledges \$2 billion to the cause at COP26, instead of viewing this as a charitable act, we should see it for what it is: repayment on a debt owed to humanity. And we might reasonably ask, as nations scramble to put together budgets to fund large-scale infrastructure upgrades, when he plans to make the rest of that repayment. Let us be clear: the digital economy has produced obscene wealth for a handful of individuals by externalizing associated environmental costs and systematically devaluing the labor that produces those profits. We should demand that this wealth be reinvested in this planet and all its inhabitants.

No Targets, No Accounting, No Plan

Without an external constraint on carbon (that, again, would be favorable to

For too long, digital technology has been able to expand without consideration of its consequences.

the computing sector), we need to start taking digital technology's role in the climate crisis seriously. Unregulated and voluntary climate pledges have been, and will continue to be, made by individual organizations in the computing sector. But those (seemingly few) pledges are often not ambitious enough to deal with the scale of the problem we are facing. Even more concerning is that pledges can be set and flaunted, without organizations being held to account to those targets. Are we willing, as a sector, to introduce our own science-based climate targets? For too long, digital technology has been able to expand without consideration of its consequences—a freedom not granted to other sectors. For example, any new Internet service can be developed and introduced with scant consideration of the societal and environmental implications, yet planning for the construction of new infrastructure (such as buildings) has oversight, and requires explicit permission. The computing sector is markedly overdue in accounting for its actions.

The End of Digital Exceptionalism

If we put aside exact percentages and look at trends, we see computing's carbon footprint growing at a rate unimaginable in other sectors.[4] In fact, climate change is seized upon as a positive use case for ever more digital solutions and an ever-increasing carbon footprint. We might call this 'digital exceptionalism'—the idea that all excesses of computing are justified because of the technology's unique capacity to increase productivity and generate profit. There are fantastic examples of computing's positive impacts, and these are often cited when rationalizing computing's privileged position in society; but the generation of profit in itself is not a guarantor of social good, particularly when economic growth comes at the cost of planetary overshoot.[9]

We seem to need reminding that computing is not exempt from having to drastically reduce emissions. Instead of assuming computing can innovate the path to a greater future, the bravest and most heroic action the computing sector could take is to show restraint and leadership, "us[ing] our knowledge and skills to advance the profession and make a positive impact" (as per ACM's mission[2]) by putting the planet above profit. It is past time for action, and the ACM community has a duty to help drive this transformation.

References

1. ACM TechBrief: Computing and Climate Change. ACM Technology Policy Council 1 (Nov. 2021).
2. About ACM: Advancing Computing as a Science & Profession. (2021); https://bit.ly/3OkXbHk
3. AI Now Institute. AI and Climate Change: How they're connected, and what we can do about it. (Nov. 2021); https://bit.ly/3xrcoAG
4. Chien, A.A. Owning computing's environmental impact. *Commun. ACM 62*, 3 (Feb. 2019), 5.
5. Crawford, K. *The Atlas of AI*. Yale University Press, 2021.
6. Dezeen. Carbon neutrality "still allows for carbon emissions" says Google sustainability lead. (July 2021); https://bit.ly/3JMjzWE
7. Freitag, C. et al. The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations. *Patterns 2*, 9 (2021), 100340.
8. Monbiot, G. Capitalism is killing the planet–It's time to stop buying into our own destruction. *The Guardian* (2021); https://bit.ly/36enyxm
9. Raworth, K. *Doughnut Economics: Seven Ways to Think Like a 21st-Century Economist*. Chelsea Green Publishing, 2017.
10. Williams, E. Environmental effects of information and communication technologies. *Nature*, 479 (2011), 354–358.

Bran Knowles (b.h.knowles1@lancaster.ac.uk) is a senior lecturer in the Data Science Institute at Lancaster University, Lancaster, U.K.

Kelly Widdicks (k.v.widdicks@lancaster.ac.uk) is a lecturer in the School of Computing and Communications at Lancaster University, Lancaster, U.K.

Gordon Blair (gblair@ceh.ac.uk) is the head of Environmental Digital Strategy at the U.K. Centre for Ecology and Hydrology, Lancaster, U.K.

Mike Berners-Lee (mike@sw-consulting.co.uk) is the founder and director of Small World Consulting and a professor in the Lancaster Environment Centre at Lancaster University, Lancaster, U.K.

Adrian Friday (a.friday@lancaster.ac.uk) is a professor in the School of Computing and Communications at Lancaster University, Lancaster, U.K.

Watch the authors discuss this work in the exclusive *Communications* video. https://cacm.acm.org/videos/climate-computings-responsibility

practice

DOI:10.1145/3486590

Article development led by acmqueue
queue.acm.org

The time is (also) way overdue for IT professional liability.

BY POUL-HENNING KAMP

The Software Industry Is *Still* the Problem

AROUND THE TIME computers were old enough to drink, software engineering guru Gerald Weinberg said: "If builders built buildings the way programmers wrote programs, then the first woodpecker that came along would destroy civilization."

This is not a plotline science fiction authors have ever neglected.

Actually, some titles are still worth a trip to the library: for example, Poul Anderson's *Sam Hall* from 1953, which shows how too much reliance on "infallible" computer surveillance can turn into an autoimmune collapse for a nation-state, or, for that matter, any large organization.

At the more obscure end of the spectrum, there is Swedish Nobel Laureate Hannes Alfvén, publishing in Swedish under the pseudonym Oluf Johannesson, with *Sagan om den stora Datamaskinen* [*Tale of the Big Computer*] from 1966.

As with almost all science fiction pieces, however, they miss the future by a wide margin. Not because they are bad at it, but because science fiction authors tend to focus on interesting and chaotic second-order effects with lots of crinkly bits around the fjords, because, let's be honest, they sell more books that way.

If any science fiction author, famous or obscure, had submitted a story where the plot was "modern IT is a bunch of crap that organized crime exploits for extortion," it would have gotten nowhere, because (A) that is just not credible, and (B) yawn!

And yet, here we are.

The good news is the ransomware attack on Colonial Pipeline in May 2021 probably marks the beginning of the end. Comforting as that might sound, it tells us very little about how that ending will turn out.

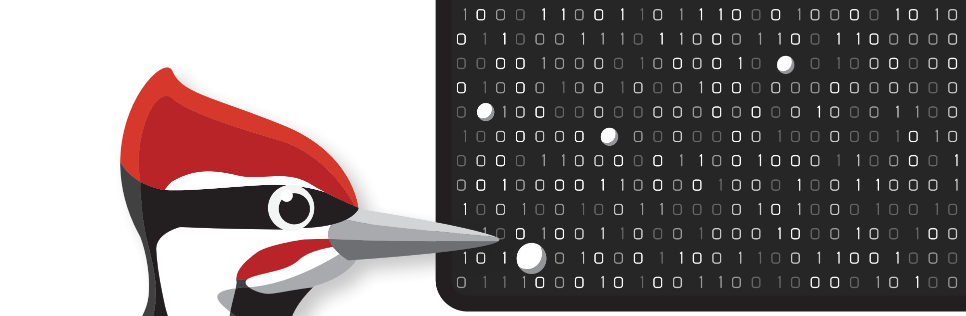
The first to react were the insurance companies. Some of them dropped the product, leaving their customers to their own devices; others were busy trying to come up with requirements and standards that would apply to their customers' claims for coverage.

I hope they put somebody competent on that assignment, because "following the industry best practice" is not going to cut it.

As I write this, 200-plus corporations, including many retail chains, have inoperative IT because extortionists found a hole in some niche, third-party software product most of us have never heard of.

Some 200 corporations are enough to argue they all "followed industry best practice," and that is precisely where Gerald Weinberg was coming from with his famous quote. The woodpecker is not leveling individual, particularly bad buildings, it is leveling civilization, because *all* the buildings are bad.

In a schoolbook instance of not seeing the forest, people who should know better are busy determining if it was a *Leuconotopicus albolarvatus*, a *Dendrocopos hyperythrus*, or some other member of *Picinae*.



Governments have finally noticed that a well-run nation-state is heavily dependent on an awful lot of computers working properly—computers that no one in the so-called "national security apparatus" previously gave much attention.

I mean, who cares how some oil company runs its billing system?

Politicians do not worry that thousands of toilets may suddenly start spewing water because goods like that have to "follow code" and adhere to product liability standards.

In contrast to this, lawyers for today's 200-plus affected corporations will find out that the full and complete extent of the third-party software vendors' liability is that they will send a new CD if the first one is unreadable.

We *badly* need real product liability for software. (See my article, "The Software Industry *IS* the Problem," in the November 2011 issue of *Communications*, p. 44.)

But what about the people who installed the toilets?

Most organizations seem to hire their first part-time IT person before they reach 20 employees. When that first IT person is replaced, the new IT person bores everybody with complaints about how "totally incompetent" the first person must have been, and, usually, is correct. When you hire somebody's teenage kid, you almost invariably get the competence you pay for.

It is also not uncommon for companies—at some point in their growth—to decide that they have to replace their entire IT installation because what they have "hampers their growth." It is only slightly less common for such projects to be a ruinous experience for all involved, because, as the eventual summary invariably goes, "Nobody seemed to know what they were doing."

Politicians do not worry about thousands of toilets spewing water because whoever installed the toilets in Colonial Pipeline's IT department had to go through examination, certification, and authorization before they were allowed to do that.

In Denmark, 129 jobs are regulated by law (https://bit.ly/33wLnPC). There are good and obvious reasons why it is illegal for any random Ken, Brian, or Dennis to install toilets or natural-gas furnaces, perform brain surgery, or certify a building is strong enough to be left outside during winter. It may be less obvious why the state cares who runs pet shops, inseminates cattle, or performs zoological taxidermy, but if you read the applicable laws, you will learn that animal welfare and protection of endangered species have many and obscure corner cases.

Notably absent, as in *totally* absent, on that list are any and all jobs related to IT; IT architecture, computers, computer networks, computer security, or protection of privacy in computer systems. People who have been legally barred and delicensed from every other possible trade—be it for incompetence, fraud, or both—are entirely free to enter the IT profession and become responsible for the IT architecture or cybersecurity of the IT system that controls nearly half the hydrocarbons to the Eastern Seaboard of the U.S.

I have not gone through the entire list, but it seems that two main requirements are: show us proof of education, which convinces us that you know what you're doing; and show us your liability insurance.

The first question is also asked in IT, but there is so much doubt about the predictive powers of the proffered certificates that many companies give candidates exams instead.

The second question is *never* asked in IT.

It is quite unusual, even for very large companies, to have an in-house licensed electrician or plumber, because, by and large, that stuff just works, just keeps working, and when it does not, you need more than a single craftsperson and suitable tools to fix it, so it is much cheaper to call in a specialized company when you need to.

With respect to gas, water, electricity, sewers, or building stability, the regulations do not care if a company is hundreds of years old or just started this morning, the rules are always the same: Stuff should just work, and only people who are licensed—because they know how to—are allowed to make it work, and they can be sued if they fail to do so.

The time is way overdue for IT engineers to be subject to professional liability, like almost every other engineering profession. Before you tell me that is impossible, please study how the very same thing happened with electricity, planes, cranes, trains, ships, automobiles, lifts, food processing, buildings, and, for that matter, driving a car.

As with software product liability, the astute reader is apt to exclaim, "This will be the end of IT as we know it!" Again, my considered response is, "Yes, please, that is precisely my point!" C

Poul-Henning Kamp spent more than a decade as one of the primary developers of the FreeBSD operating system before creating the Varnish HTTP Cache software, which approximately one-fifth of all Web traffic goes through. He is an independent contractor; one of his recent projects was a supercomputer cluster to stop the stars twinkling in the mirrors of European Southern Observatory's new Extremely Large Telescope (ELT).

practice

DOI:10.1145/3498578

Article development led by acmqueue
queue.acm.org

Software engineering teams can exploit attackers' human nature by building deception environments.

BY KELLY SHORTRIDGE AND RYAN PETRICH

Lamboozling Attackers: A New Generation of Deception

DECEPTION IS A powerful resilience tactic that provides observability into attack operations, deflects impact from production systems, and advises resilient system design. A lucid understanding of the goals, constraints, and design trade-offs of deception systems could give leaders and engineers in software development, architecture, and operations a new tactic for building more resilient systems—and for bamboozling attackers.

Unfortunately, innovation in deception has languished for nearly a decade because of its exclusive ownership by information security specialists. Mimicry of individual system components remains the status-quo deception mechanism despite growing stale and unconvincing to attackers, who thrive on interconnections between components and expect to encounter *systems*. Consequently, attackers remain unchallenged and undeterred.

This wasted potential motivated our design of a new generation of deception systems, called *deception environments*. These are isolated replica environments containing complete, active systems that exist to attract, mislead, and observe attackers. By harnessing modern infrastructure and systems design expertise, software engineering teams can use deception tactics that are largely inaccessible to security specialists. To help software engineers and architects evaluate deception systems through the lens of systems design, we developed a set of design principles summarized as a pragmatic framework. This framework, called the *FIC trilemma*, captures the most important dimensions of designing deception systems: fidelity, isolation, and cost.

The goal of this article is to educate software leaders, engineers, and architects on the potential of deception for systems resilience and the practical considerations for building deception environments. By examining the inadequacy and stagnancy of historical deception efforts by the information security community, the article also demonstrates why engineering teams are now poised—with support from advancements in computing—to become significantly more successful owners of deception systems.

Deception: Exploiting Attacker Brains

In the presence of humans (attackers) whose objectives are met by accessing, destabilizing, stealing, or otherwise leveraging other humans' computers without consent, software engineers must understand and anticipate this type of negative shock to the systems they develop and operate. Doing so

IMAGE BY SLAWOMIR ZELASKO

involves building the capability to collect relevant information about attackers and to implement anticipatory mechanisms that impede the success of their operations. Deception offers software engineering teams a strategic path to achieve both outcomes on a sustained basis.

Sustaining resilience in any complex system requires the capacity to implement feedback loops and continually learn from them. Deception can support this continuing learning capacity. The value of collecting data about the interaction between attackers and systems, which we refer to as *attack observability*, is generally presumed to be the concern of information security specialists alone. This is a mistake. Attacker effectiveness and systems resilience are antithetical; one inherently erodes the other. Understanding how attackers make decisions allows software engineers to exploit the attackers' brains for improved resilience.

Attack observability. The importance of collecting information on how attackers make decisions in real operations is conceptually similar to the importance of observability and tracing in understanding how a system or application *actually* behaves rather than how it is *believed* to behave. Software engineers can attempt to predict how a system will behave in production, but its actual behavior is quite likely to deviate from expectations. Similarly, software engineers may have beliefs about attacker behavior, but observing and tracing *actual* attacker behavior will generate the insight necessary to improve system design against unwanted activity.

Understanding attacker behavior starts with understanding how humans generally learn and make decisions. Humans learn from both immediate and repeated interactions with their reality (that is, experiences). When making decisions, humans supplement preexisting knowledge and beliefs with relevant experience accumulated from prior decisions and their consequences. Taken together, human learning and decision-making are tightly coupled systems. Given that attackers are human beings—and even automated attack programs and platforms are designed by humans—this tight coupling can be leveraged to destabilize attacker cognition.

In any interaction rife with conflict, such as attackers vs. systems operators, information asymmetry leads to core advantages that can tip success toward a particular side. Imperfect information means players may not observe or know all moves made during the game. Incomplete information means players may be unaware of their opponents' characteristics such as priorities, goals, risk tolerance, and resource constraints. If one player has more or better information related to the game than their opponent, this reflects an information asymmetry.

Attackers choose an attack plan based on preexisting beliefs and knowledge learned through experience about operators' current infrastructure and protection of it.[1] Operators choose a defense plan based on preexisting and learned knowledge about attackers' beliefs and methods.

This dynamic presents an opportunity for software engineers to use deception to amplify information asymmetries in their favor.[7] By manipulating the experiences attackers receive, any knowledge gained from those experiences is unreliable and will poison the attackers' learning process, thereby disrupting their decision-making.

Deception systems allow software engineers to exacerbate information asymmetries in two dimensions: exposing real-world data on attackers' thought processes (increasing the value of information for operators); and manipulating information to disrupt attackers' abilities to learn and make decisions (reducing the value of information for attackers).

The rest of the article will discuss the challenges and potential of deception systems to achieve these goals in real-world contexts.

Conventional deception approaches are unconvincing to attackers with a modicum of experience.

The History of Honeypots

The art of deception has been constrained by information security's exclusive ownership of it. The prevailing mechanism used to implement deception is through a host set up for the sole purpose of detecting, observing, or misdirecting attack behavior, so that any access or usage indicates suspicious activity. These systems are referred to as honeypots in the information security community. It is worth enumerating existing types of *honeypots* to understand their deficiencies.

Levels of interactivity. Honeypots are typically characterized by whether they involve a low, medium, or high level of interactivity.

Low interaction (LI) honeypots are the equivalent of cardboard-cutout decoys; attackers cannot interact with them in any meaningful way. LI honeypots represent simple mimicry of a system's availability and are generally used to detect the prevalence of port scanning and other basic methods attackers use to gather knowledge relevant for gaining access (somewhat like lead generation). They may imitate a specific port or vulnerability and record successful or attempted connections.

Medium interaction (MI) honeypots imitate a specific kind of system, such as a mail server, in enough depth to encourage attackers to exploit well-known vulnerabilities, but they lack sufficient depth to imitate full system operation. Upon an exploitation attempt, MI honeypots send an alert or record the attempt and reject it. They are best for studying large-scale exploitation trends of public vulnerabilities or for operating inside of a production network where any access attempt indicates an attack in progress.

High interaction (HI) honeypots are vulnerable copies of services meant to tempt attackers, who can exploit the service, gain access, and interact with the base operating-system components as they normally would. It is uncommon for HI honeypots to include other components that imitate a real system. For the few that do, it is usually a side effect of being built by transplantation. HI honeypots usually send an alert upon detection of an attacker's presence, such as after successful exploitation of the vulnerable software.

Limitations of honeypots. While LI

and MI honeypots are generally understood to be ineffectual at deceiving attackers[9] (and thus can be dismissed as applicable options for real-world deception), the existing corpus of HI honeypots is primitive as well. Conventional deception approaches are unconvincing to attackers with a modicum of experience. Attackers need only ask simple questions—Does the system feel real? Does it lack activity? Is it old and forgotten?—to dissipate the mirage of HI honeypots.

The limitations of HI honeypots mean that attackers often uncover their deceptive nature by accident. HI honeypots also lack the regular flow of user traffic and associated wear of production systems—a dead giveaway for cautious attackers.

Finally, a fundamental flaw of all honeypots is that they are built and operated by information security specialists, who are typically not involved in software architecture and are largely divorced from software delivery. They may know at a high level how systems are supposed to behave but are often unaware of the complex interactions between components that are pivotal to systems function. As such, this exclusive ownership by security specialists represents a significant downside to current deception efficacy.

Modern Computing Enables New Deception

A new generation of deception is not only possible, but also desirable given its strategic potential for systems resilience. The design and ownership of this new category, deception environments, reflects a significant departure from the prior generation. Deception environments are sufficiently evolved from honeypots that they represent a new, distinct category.

It is not surprising that attackers find individual honeypot instances unconvincing, given their expertise in attacking systems and understanding the interrelation between components to inform their operations. The combination of new types of computing and ownership by software engineers means that environments dedicated to distributed deception can be created that more closely resemble the types of systems attackers expect to encounter.

The goal of traditional honeypots is to determine how frequently attackers are using scanning tools or exploiting known vulnerabilities; tracing the finer nuances of attacker behavior or uncovering their latest methodology is absent from deception projects to date. Deception environments serve as a means to observe and understand attacker behavior throughout all operational stages and as platforms for conducting experiments on attackers capable of evading variegated defensive measures. This concentrates efforts on designing more resilient systems and makes fruitful use of finite engineering attention and resources.

A few dimensions of modern infrastructure are pivotal in nurturing a new deception paradigm with lower costs and more efficacious design.

- **Cloud computing.** The accessibility of cloud computing enables the ability to provision fully isolated infrastructure with little expense.
- **Deployment automation.** Full systems deployment automation and the practice of defining infrastructure declaratively, commonly referred to as IaC (infrastructure as code), decreases operational overhead in deploying and maintaining shadow copies or variants of infrastructure.
- **Virtualization advancements.** The widespread availability of nested virtualization and mature, hardened virtualization technologies inspires confidence that attackers are isolated from production, makes it possible to observe them in more detail, and extracts extra density out of computing resources.
- **Software-defined network (SDN) proliferation.** With the ability to define networks programmatically, isolated network topology dedicated to attackers can be created without incurring additional cost.

New ownership. This is another crucial catalyst for this latest generation of deception. Ownership based on systems design expertise, rather than security expertise, creates the dynamism necessary for deception systems to succeed against similarly dynamic opponents.

Software engineering teams are already executing the necessary practices. Software operators can repurpose their unique system deployment templates for building production environments and variants (such as staging environments) toward building powerful deception systems. They can then derive attack data that is distinctly applicable to their environments and cannot be garnered elsewhere. As a result, software engineers are more qualified for the endeavor than security teams and can gain a highly effective observability tool by deploying deception environments.

Designing Deception Environments

The design philosophy underlying deception environments is grounded in repurposing the design, assets, and deployment templates of a real system instead of building a separate design for deception (as is the status quo). Deception becomes a new environment generated at the end of software delivery pipelines after development, staging, pre-production, and production. From this foundation, attacker skepticism can be preempted by designing a deception environment that feels "lived in" through tactics such as replaying traffic and other methods of simulating system activity.

Starting with the design of a genuine production system provides an inherent level of realism to bamboozle attackers and glean insights pertinent to refining resilience in the real system. Since every system has different resilience concerns, this also offers an opportune and safe test of how tactics perform against real attackers in a pseudo-real environment.

The FIC trilemma. Traditional honeypot design has focused on initial access, and success is determined by how well a honeypot can mimic the outer shape of a system. This framing limits the ability to evaluate approaches beyond the rudimentary ones seen to date.

The new model proposed here evaluates deception systems along three axes: fidelity, isolation, and cost (See Figure 1 and the sidebar), representing a trilemma: The three properties are generally in conflict and therefore cannot be fully achieved simultaneously. Understanding the FIC trilemma—and the trade-offs between each of its axes—is vital for designing successful deception environments.

Fidelity refers to the deception system's credibility to attackers and its ability to support attack observability.

The FIC Trilemma Sweet Spots for Deception Environments

FIC—The most important dimensions of designing deception systems: Fidelity, Isolation, and Cost.

Replicomb—A full replica of a production host with imitated load and a purposefully vulnerable component.

Honeyhive—A full, scaled-down replica of an entire production environment with activity flowing through a web of replicomb hosts.

A credible deception system is effective at deceiving attackers into thinking the system is real; it avoids falling into the "uncanny valley." Attackers often interrogate compromised systems to unmask mirages and avoid revealing their methods. Attackers expect certain basic traits in systems, such as running a service, receiving production-like traffic, connecting to the wider Internet, coordinating with other services over a local network, being orchestrated and monitored by another system, and not having traces of debuggers or other instrumentation tools.

A highly credible deception system will provide sufficient depth to stimulate extended attacker activity, luring even cautious attackers into moving between hosts and revealing their methods across the attack delivery life cycle. This begets a detailed and high-quality record of behavior for engineers to gain an accurate understanding of attacker decision-making. Greater accuracy and depth in extracting and recording activity informs better system design that makes future iterations more resilient to attack.

Isolation refers to the degree to which a deception system is *isolated* from the real environment or data, and is the second axis of the FIC trilemma. Operators are loath to jeopardize the availability of the real system or data privacy in order to learn about attacker behavior. A secondary element is the ability to keep attackers isolated from each other. This permits study of each attacker's behavior independently.

Cost refers to the computing infrastructure and operational overheads required to deploy and maintain deception systems. As computing expenses continue to plummet, cost shifts to operational burden—which should not be underestimated. Expensive deception systems are unlikely to be fully deployed or maintained and will thereby fail to serve their purpose.

Mapping different types of deception systems to points around the trilemma elucidates the value of this model. As a starting point, let us consider which types of systems reflect extreme realizations of each of these axes: perfect fidelity, total isolation, and maximum cost (as shown in Figure 2).

Real production systems reside at the intersection of *perfect fidelity, little cost, and no isolation*. These systems are likely to encounter attackers and be monitored by operators, because production is where organizations realize value from software-development activity (that is, it makes the money, which attackers and organizations similarly appreciate).

In contrast, LI honeypots reside at the intersection of *no fidelity, little cost, and perfect isolation*. They gather limited information about attackers and present them with a transparent trick; however, they are easy to deploy, can detect broad attack trends, and offer a risk-free incident impact.

At the intersection of *perfect fidelity*, *full isolation*, and *maximum cost* resides a hypothetical datacenter dedicated to deception. As an example, imagine a complete copy of a production datacenter with identical monitoring and maintenance, as well as perfect simulation of real traffic using an army of distributed clients. This obviously bears an exorbitant cost in terms of design and operation but provides high fidelity and full isolation.

To explore the FIC trilemma further, Figure 3 evaluates the aforementioned approaches from the information security community.

MI honeypots offer minimal supplemental fidelity and cost about the same to deploy as LI honeypots; hence, they occupy a space close to LI honeypots. HI honeypots represent a minor increase in fidelity, at some cost, but are unable to fool most attackers. Even when simulated load is applied to boost authenticity, HI honeypots still suffer from the limitations of imitative design rather than sharing lineage with real existing systems.

Sweet spots for deception environments. The model for deception environments supports solutions in previously unexplored spaces in the trilemma (see Figure 4). The following two trilemma "sweet spots" provide mechanisms for uncovering a richer and higher volume of attacker behavior for advanced observability.

Figure 1. The FIC trilemma for deception systems.

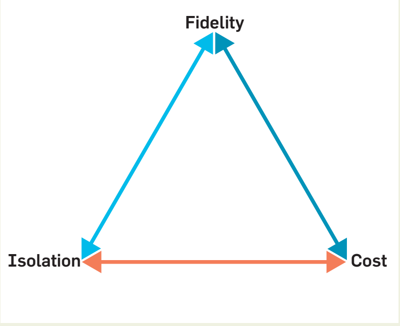


Figure 2. Example deception systems mapped to the FIC trilemma.

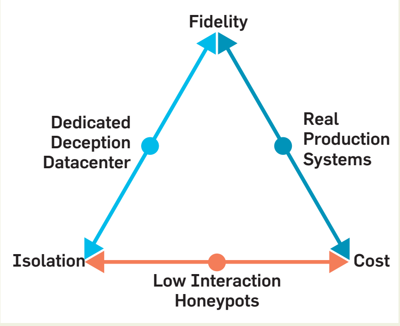
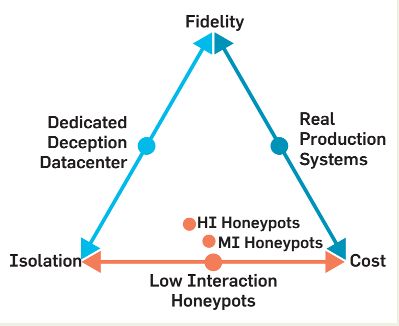


Figure 3. MI and HI honeypots on the trilemma.



Systems in the first sweet spot, called *replicombs*, are downgraded replicas of production hosts that run with the same set of monitoring, orchestration, and supporting services deployed in real production environments. The replica is fed with simulated or replayed load from actual systems. A full replica host with a production-like load creates a deception system that, to an attacker, appears indistinguishable from a real host (as illustrated in Figure 5).

Modern deployment practices such as IaC make crafting downgraded replicas easier, and the plummeting cost of cloud computing makes deployment of sizable systems inexpensive. While still higher cost than a honeypot, a replicomb offers palpable enhancements: It features impressive fidelity and supports inspecting an expansive range of attacker behavior beyond initial access. Because of this, the replicomb occupies a space on the trilemma closer to the full datacenter replica. This is a sweet spot because it should, if implemented correctly, appear to be a real individual production host to even a cautious, skeptical attacker.

Systems in the second proposed sweet spot, called *honeyhives*, extend the replicomb approach with a full network of like-production hosts to observe how attackers move from their initial point of access onto adjacent hosts and services. Complete but scaled-down copies of an entire environment are deployed as a honeyhive with simulated, replayed, or mirrored activity flowing through the entire system. Therefore, a honeyhive yields a thoroughly lifelike environment for observing and conducting experiments on attackers, even if their behavior spans multiple systems (see Figure 6).

The honeyhive environment may sound similar to a preproduction or staging environment—and it is. Modern IaC practices and inexpensive full isolation via cloud computing allow for a deception system such as a honeyhive to be deployed at a more reasonable cost than previously feasible. The honeyhive occupies a space on the trilemma nearest to the full datacenter replica, offering profound fidelity to attackers and in the intelligence gathered from it. With a honeyhive, behavior can be observed through more stages of an attacker's operation.

A replicomb is the starting point for

Figure 4. The FIC Sweet Spot: Honeyhives and replicombs.

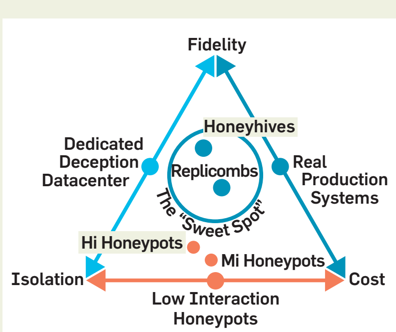


Figure 5. An example of replicomb deployment.

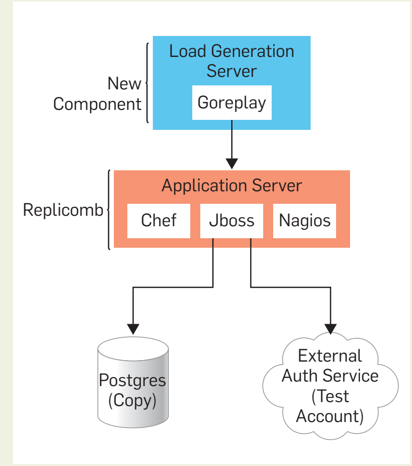


Figure 6. Example honeyhive based on a production environment.

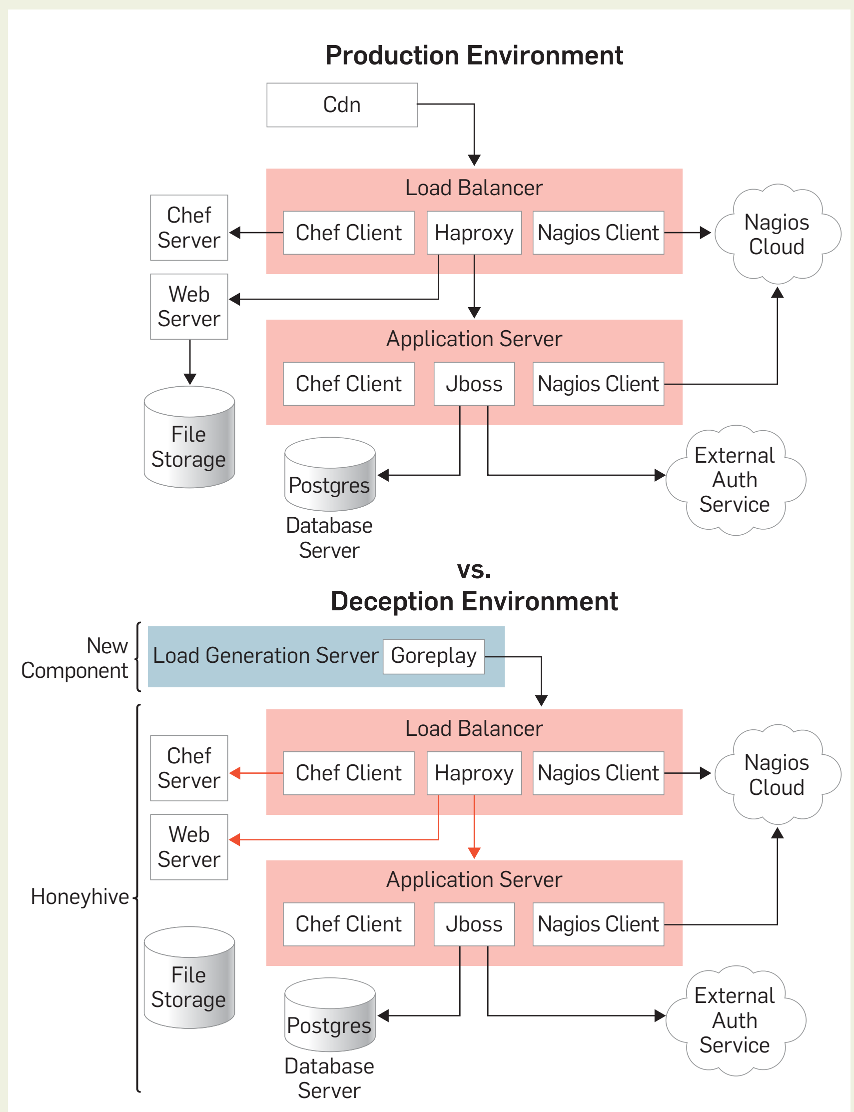
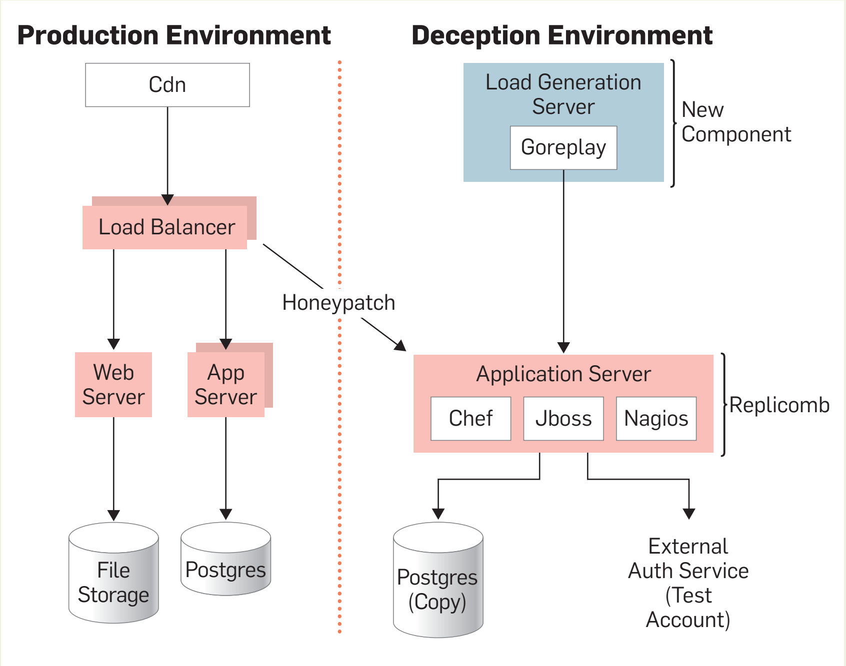


Figure 7. An example replicomb environment with honeypatching.



a honeyhive, but outsized fidelity is unlocked by deploying the rest of the environment. A replicomb is effectively a copy of a service, so it requires simulated load to appear real. A honeyhive, in contrast, needs only simulated load applied to any points that would naturally interact with users; only one "true" replicomb is required as the initial entry point. The other replicomb hosts receive traffic from their peers just as they would in a production environment, so the honeyhive simply needs some external traffic to engender realistic flows.

Real-world implementation. Building replicombs and honeyhives is no more difficult for software organizations than setting up a new variant of an existing environment tier through IaC declarations. Deploying a replicomb is similar to a canary release of the chosen service, and deploying a honeyhive is similar to a soak or load test environment.

With this said, safely building deception environments requires careful attention to details beyond the usual concerns when creating a new environment.

Isolation boundaries. Where does the isolation boundary exist between the deception environment and any other environments that process user data or must remain available? How permeable is this boundary? Purposefully deploying vulnerable instances of a service without properly isolating them from user traffic is dangerous.

Similarly, some organizations run multiple environments within the same network, allowing direct communication between them. Deception environments should not follow this pattern but instead should be deployed with little to no ability to communicate with environments dedicated to other purposes, especially those handling critical production traffic. Virtualization techniques, SDNs and cloud computing can be used to create fully isolated networks for deception environments.

Discoverability. Attackers must be able to discover deception environments to collect real data on their attack operations. Placing a deception environment on a public IP address without any association to your organization attracts only the attackers searching across the Internet for open holes to poke. Placing a deception system inside a production environment as a discoverable host captures behavior only after attackers are already inside and seeking additional hosts.

One technique to trap attackers seeking access to a specific organization and then to observe their behavior across all stages of their operation is *honeypatching*. This technique directs traffic intended to exploit a known and already-patched vulnerability in production to the deception environment's unpatched service by leveraging the configurability of modern firewalls and load balancers (see Figure 7).

Tamper-free observation. A salient benefit of funneling attacks into a deception environment is the ability to trace the attackers' actions without any risk to actual service availability. This tracing should be invisible to attackers and resistant to tampering by them. Once attackers gain access to a deception environment, they can potentially manipulate any monitoring or observability tools running within it. Accordingly, the best way to ensure tamper-free observation of attacker behavior is to deploy these tools outside the environment but peering inward.

Network behavior can be observed by capturing all packets entering, leaving, and moving between hosts by using a CSP's (cloud service provider) native features for archiving traffic within a virtual network or by using the packet-capture facilities of virtualization systems. Host behavior can be observed by taking regular snapshots of memory and disk to view the deception system's exact state at a given time.

To improve resilience against attackers evading any monitoring of their actions, packet capture and periodic snapshots can be supplemented with standard observability tools. Essential events such as process launch and file activity collected inside the environment can also enrich the trace of attacker activity.

Accidental data exposure. Organizations may inadvertently accept liability by purposefully exposing user data to attackers in the deception environment. This problem can be mitigated by anonymizing or scrambling traffic before it is replayed into the deception environment.

Generating synthetic datasets—those that mimic production data but do not include any real user data—is an existing approach for populating preproduction, staging, and other test environments while still complying with privacy regulations (such as HIPAA). Organizations in less privacy-conscious industries may need to adopt a similar approach for deception environments to avoid unwanted liability.

Ownership. The conventional view of deception systems is that they reside in the domain of information security. With modern advancements in deployment tooling and methodology, creating variants of production systems is a straightforward exercise—and deception environments are simply another variant of the system. Software engineers can consequently deploy and maintain effective deceptions in a more straightforward, predictable, low-effort, automated, consistent, and understandable way.

Security expertise is not a prerequisite for developing and operating deception environments. In fact, it often constrains and impairs judgment of strategic options. Engineering teams naturally gravitate toward improvements to design or workflows instead of relying on status-quo "best" practices seldom informed by systems thinking. Attackers think in systems; they develop systems to achieve their objectives and incorporate feedback during operations. By treating an attacker as a kindred engineer with the exact opposite goals to yours, your mindset will be amply authentic and constructive to wield and benefit from deception environments.

Harvesting Deception's Full Potential

Here are a few powerful use cases to harvest the potential of a deception environment after it is deployed.

Resilient system design. The data generated from replicombs and honeyhives can inform more resilient system design. Minimizing the time to detect and respond to destructive activity that precipitates service downtime is correlated with organizational performance[2]—and attacks are firmly in the category of such ruinous activity. A dedicated sandbox for exploring how attacks impact systems is an invaluable tool for anticipating how production systems will behave when failure occurs and preempting it through design improvements.

Attackers will interact with monitoring, logging, alerting, failover, and service components in ways that stress their overall reliability. A resilient system must be aware of and recover from failures in any of these components to preserve availability. Deception environments can corroborate any measures implemented to support visibility into and recovery from component failure.

Deception environments can also expose opportunities for architectural improvement in operability and simplicity. For example, if spawning remote interactive shells (so attackers can write their tools to disk) is a consistent attacker behavior seen across deception environments, this evidence could motivate a design specification of host immutability to eliminate this option for attackers.[5]

Importantly, this aligns with a future in which product and engineering teams are accountable for the resilience of the systems they develop and operate (including resilience to attacks).[4] In the spirit of security chaos engineering (SCE), software engineers, architects, site reliability engineers, and other stakeholders can leverage a feedback loop fueled by real-world evidence—such as that produced by replicombs and honeyhives—to inform improvements across the software-delivery life cycle.

Attacker tracing. Deception environments equip software engineers, architects, and other systems practitioners to "trace" the actions of attackers. Attack observability enables pragmatic threat modeling during design and planning without security expertise as a prerequisite. Since attacker behavior is traced in detail on a system with the same shape as a real system, the resulting insight is perfect for modeling likely decision patterns via frameworks such as attack trees.[6] It can inform revisions to systems design, adjustments to monitoring and observability, or revised resilience measures.

Attack trees are a form of decision tree that graphs decision flows—how attackers will take one path or another in a system to reach their objectives. While a safe default assumption is that attackers will pursue the lower-cost decision path, the in-the-wild evidence collected from deception environments can validate or update existing hypotheses about how attackers learn and make decisions in specific systems. For example, attacker tracing can establish which tactics (or combinations of them) nudge attackers toward certain choices.

This elucidation of behavioral patterns across the attack life cycle can be visualized as different branches on the attack tree and aid in prioritizing system design changes. (An example tool for visualizing security decision trees is Deciduous, an open source web app created by the authors and available at https://www.deciduous.app/). It can also excavate the hidden flows within systems that are ordinarily discovered only upon failure and system instability. Attackers are adept at ferreting out unaccounted flows to achieve their objectives, so tracing their traversal paints a more precise picture of the system.

Attacker tracing can also inform experimentation; each branch on the attack tree represents a chain of hypotheses that elicit specific experiments. Attacker tracing could also extract thorough characterizations of attackers—their objectives, learning ability, level of risk aversion, degree of skepticism, and other behavioral factors. Such characterization could galvanize personalized deception by combining conditional logic and a "terraforming" approach.

Experimentation platform. A lifelike environment indistinguishable from a production environment maximizes success in deceiving attackers across all levels of capability. A deception environment can therefore serve as a platform for conducting experiments to test hypotheses on how attackers will behave in various circumstances. Attacker tracing—especially when accompanied by attack trees—can directly inform hypotheses for experiments.

Solution efficacy. Experimentation can test the efficacy of monitoring or resilience measures and whether they can be subverted without the operator's knowledge. For example, deception environments can reveal how attackers might respond to architecture redesigns or to substitutions of infrastructure components (swapping for an equivalent capability, as discussed in the next section).

Similar to validating system performance with load or soak tests, it is valuable to validate system resilience under various failure scenarios (including exposure to attackers). SCE rests on this foundation of experimentation; fault injection generates evidence that builds knowledge of systems-level dynamics—informing continuous improvement of systems resilience and creating muscle memory for incident response.[4] Through this lens, deception environments become an experimentation tool in the SCE arsenal.

Fidelity thresholds. Fidelity degradation experiments can divulge how attackers react to environments with varying levels of fidelity—uncovering the point at which a system begins to look like an "uncanny valley" to different attackers. Removing components one by one (similar to A/B testing) can surface which aspects of the environment attackers use to evaluate realism.

For example, attackers may treat systems running without monitoring tools as insufficiently important to bother ransoming, since system criticality influences the victim's willingness to pay. Conducting experiments by alternatively disabling and enabling monitoring and logging subsystems can unveil to what extent attackers will flee from unmonitored systems.

For well-resourced attackers capable of gaining access to both the honeyhive and the production environment, swapping standard components for substitutes can disrupt their attack plans. These substitutes expose the same interface and perform the same function (similar to Coke vs. Pepsi), but the difference in brand name introduces unreliability into the attacker's operational knowledge. Swapping components and testing system behavior under simulated load is common in engineering disciplines and useful for evaluating many categories of hypotheses (such as whether a new component performs better or is easier to make operational).

Access difficulty. The difficulty involved in accessing the deception environment can be tuned to study different types of attackers and their perceptions of fidelity. Since attackers expect certain victims to have a basic level of software-patching hygiene, advertising trivially exploitable vulnerabilities can degrade a deception's fidelity. By selecting which vulnerabilities to telegraph as patched or unpatched (that is, honeypatches), the accessibility of initial entry points can be adjusted to meet attacker expectations.

Building replicombs and honeyhives is no more difficult for software organizations than setting up a new variant of an existing environment tier through IaC declarations.

Honeytokens for flavor. Augmenting honeyhives with other deception techniques can measure the efficacy of those techniques and trace an attacker's progress in more detail. For example, deploying cloud honeytokens throughout the environment can warn operators when attackers gain access to various systems and to what extent they attempt to access cloud resources. (An example of cloud honeytokens is the AWS [Amazon Web Services] key canarytoken by Thinkst, available for free at https://canarytokens.org/generate.)

Future Opportunities

The potential use cases for deception environments described thus far can be realized with contemporary practices and tools. From here, modest extensions to the underlying technology can serve as general-purpose tools that can even benefit disciplines beyond deception.

Just-in-time terraforming. Modern virtualization could support just-in-time creation of isolated deception virtual machines (VMs) via copy-on-write or page deduplication. (This was proven possible in 2005 but never adopted elsewhere.)[8] This process is similar to how operating systems employ lazy copying of memory pages after processes fork. This could reduce costs by sharing resources among deception environments and creating them only when an attacker first gains access.

A "systems terraforming" approach could even render the illusion of an entire network of hosts that are reified only when an attacker attempts to connect to them. Cloud networking and hypervisor layers already cooperate to route network traffic within a VPC (virtual private cloud) to the physical hardware associated with the intended instance. For lower overhead of unused infrastructure, these layers could instead treat VM instances similarly to serverless functions: powering on once they receive traffic but otherwise remaining suspended, hibernated, powered off, or paged out to disk.

Clever improvements to the network and hypervisor layers could facilitate this freezing of idle services, hosts, and infrastructure. These assets would be unfrozen upon first contact over the network and have their execution fast-forwarded to the point of interactivity. Once instances finish processing incoming traffic and return to idle states, they would be put back into deep freezes to reduce resource usage. This approach could reduce the cost of blue-green deployments, preproduction environments, and other scenarios where infrastructure mostly idles.

Instance emulation. Advancements in virtualization technology that would better emulate the proprietary hardware and local instance metadata endpoints of AWS and Google Cloud Platform (GCP) would allow creation of deception environments that appear to be real Amazon Elastic Compute Cloud (EC2) and GCP instances. Full emulation of CSP APIs could lead to benefits such as offline testing of cloud environments, higher-density testing of an entire fleet on a single host, and fully isolated honeyhives that live on a single machine.

Scalable honeypatching. Networking technologies such as content delivery networks (CDNs), routers, load balancers, service meshes, or web application firewalls can be reconfigured to support low-effort honeypatching at scale. Rather than blocking exploitation attempts, attackers could be trivially redirected to a deception environment (with the sole new overhead of specifying where to direct suspicious traffic). Additionally, vulnerability signatures are currently treated as post-hoc security mechanisms but could be distributed alongside software updates instead for straightforward and swift implementation. If supplied to the aforementioned networking technologies, signatures could permit pattern matching in data or protocol streams.

Anonymization via mirroring. Traffic-mirroring technologies, such as those integrated into service meshes and VPCs, could be extended to include data anonymization features that operate at the application protocol layer. Current anonymization techniques operate at the packet layer,[3] which is insufficient to anonymize user data being mirrored into a deception environment. Offline data anonymization techniques such as privacy-preserving encryption and pseudonymization could be integrated with traffic-mirroring systems to uphold user privacy and meet compliance requirements while replicating a completely natural traffic pattern in a deception environment.

Hypervisor-based observability. Tracing and observability are core user requirements of modern server operating systems but are often simple to subvert. Such tools commonly execute at the same level of privilege as the workloads they monitor; everything is root. Ideally, installing and running a userland agent should not be necessary to get basic metrics and telemetry out of the kernel for systems monitoring.

Toward this goal, operating systems could expose core system events such as process and file operations directly to hypervisors over a common protocol. This would surface visibility that could not be subverted by an attacker operating inside the VM and would prevent observability outages provoked by resource exhaustion (as can often occur on overloaded hosts).

Burstable memory usage. The infrastructure cost of deception environments could be further reduced if CSPs supported traditional virtualization features such as ballooning and compressed or swapped memory. CSPs could then offer burstable performance instances featuring the ability to burst memory usage to a higher level when required (while offering lower baseline performance and therefore lower cost). This is similar to AWS's existing credits-based system for workloads that require infrequent bursts of CPU usage or GCP's static customization of memory assignment.

With advanced hypervisor extensions, CSPs could migrate VMs across physical instances when their activity bursts and they require more resources. Instead of stopping idle instances on a shared physical machine when one instance bursts, idle instances could be temporarily migrated to another host machine or swapped to disk. This approach is possible with current technology but is not yet implemented by CSPs.

Per-account billing limits. To restrict the amount of money attackers can spend on your behalf, per-account billing limits are stronger than billing alerts. Unfortunately, CSPs do not provide tools to limit spending by account or project and alert only when unusual activity occurs or thresholds are exceeded. These tools are adequate when availability eclipses cost concerns but are incapable of enforcing a true backstop for resource consumption. CSPs have effective tools for isolating every resource except for their customers' wallets; customers could ask them to add this capability.

Parting Thoughts

Imagine a world in which developers and operators of systems exploit attackers as much as attackers exploit defenders. By leveraging system-design knowledge and modern computing to deploy deception environments, software engineering teams can successfully bamboozle attackers for fun and profit while deepening systems resilience.

References

1. Alderson, D.L., Brown, G.G., Carlyle, W.M., Wood, R.K. Solving defender-attacker-defender models for infrastructure defense. Center for Infrastructure Defense, Operations Research Department, Naval Postgraduate School, Monterey, CA, 2011; https://calhoun.nps.edu/handle/10945/36936.
2. Forsgren, N., Humble, J., Kim, G. *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations.* IT Revolution, 2018.
3. Hyojoon, K., Chen, X., Brassil, J., Rexford, J. Experience-driven research on programmable networks. *ACM SIGCOMM Computer Commun. Rev. 51*, 1 (2021), 10–17; https://dl.acm.org/doi/10.1145/3457175.3457178.
4. Rinehart, A., Shortridge, K. *Security Chaos Engineering.* O'Reilly Media, 2020.
5. Shortridge, K., Forsgren, N. Controlled chaos: The inevitable marriage of DevOps & security. Presentation at 2019 Black Hat USA; https://bit.ly/3sMZuZI.
6. Shortridge, K. The scientific method: security chaos experimentation & attacker math. Presentation at RSA 2021 Conf.; https://bit.ly/3LJVBxp.
7. Veksler, V.D., Buchler, N., LaFleur, C.G., Yu, M.S., Lebiere, C., Gonzalez, C. Cognitive models in cybersecurity: Learning from expert analysts and predicting attacker behavior. *Frontiers in Psychology 11*, 1049 (2020); https://www.frontiersin.org/articles/10.3389/fpsyg.2020.01049/full.
8. Vrable, M., Ma, J., Chen, J., Moore, D., Vandekieft, E., Snoeren, A.C., Voelker, G.M., Savage, S. Scalability, fidelity, and containment in the Potemkin virtual honeyfarm. *ACM SIGOPS Operating Systems Rev. 39*, 5 (2005), 148–162; https://dl.acm.org/doi/10.1145/1095809.1095825.
9. Zhang, L., Thing, V.L.L. Three decades of deception techniques in active cyber defense—retrospect and outlook. *Computers & Security 106*, 102288 (2021); https://bit.ly/36g14LT7.

Kelly Shortridge is a senior principal in product technology at Fastly, co-author with Aaron Rinehart of *Security Chaos Engineering* (O'Reilly Media), and is an expert in resilience-based strategies for systems defense. Their research on applying behavioral economics and DevOps principles to information security has been featured in top industry publications and is used to guide modernization of information security strategy globally.

Ryan Petrich is an SVP at a financial services company and was previously chief technology officer at Capsule8. Their current research focuses on using systems in unexpected ways for optimum performance and subterfuge. Their work spans designing developer tooling, developing foundational jailbreak tweaks, architecting resilient distributed systems, and experimenting with compilers, state replication, and instruction sets.

DOI:10.1145/3486897

Standardizing computational reuse and portability with the Common Workflow Language.

BY MICHAEL R. CRUSOE, SANNE ABELN, ALEXANDRU IOSUP, PETER AMSTUTZ, JOHN CHILTON, NEBOJŠA TIJANIĆ, HERVÉ MÉNAGER, STIAN SOILAND-REYES, BOGDAN GAVRILOVIĆ, CAROLE GOBLE, AND THE CWL COMMUNITY

Methods Included

COMPUTATIONAL WORKFLOWS ARE widely used in data analysis, enabling innovation and decision-making for the modern society. But their growing popularity is also a cause for concern. Unless we standardize computational reuse and portability, the use of workflows may end up hampering collaboration. How can we enjoy the common benefits of computational workflows and eliminate such risks?

To answer this general question, in this work we advocate for workflow thinking as a shared method of reasoning across all domains and practitioners, introduce Common Workflow Language (CWL) as a pragmatic set of standards for describing and sharing computational workflows, and discuss the principles around which these standards have become central to a diverse community of users across multiple fields in science and engineering. This article focuses on an overview of CWL standards and the CWL project and is complemented by the technical detail available in the CWL standards.[a]

Workflow thinking is a form of "conceptualizing processes as recipes and protocols, structured as dataflow [or workflow] graphs with computational steps, and subsequently developing tools and approaches for formalizing, analyzing, and communicating these process descriptions."[14] It introduces the workflow, an abstraction which helps decouple expertise in a specific domain—for example, specific science or engineering fields—from computing expertise. Derived from workflow thinking, a *computational workflow* describes a process for computing where different parts of the process (the tasks) are interdependent—for instance, a task can start processing after its predecessors have (partially) completed and where data flows between tasks.

Currently, many competing systems exist to enable simple workflow execution (*workflow runners*) or offer comprehensive management of workflows and data (*workflow management systems*). Each has its own syntax or method for describing workflows and infrastructure requirements, which can limit computational reuse and portability. Although dataflows are becoming more complex, most workflow abstractions do not enable explicit specifications of dataflows, significantly increasing the cost to have third parties reuse and port the workflow.

a Common Workflow Language Standards, v1.2: https://w3id.org/cwl/v1.2/.

» key insights

- **Common Workflow Language is a set of open standards for describing and sharing computational workflows, used in many science and engineering domains.**
- **CWL standards support critical workflow concepts such as automation, scalability, abstraction, provenance, portability, and reusability.**
- **CWL standards are developed around core principles of community and shared decision making, reuse, and zero cost for participants.**

We thus identify an important problem in the broad, practical adoption of workflow thinking: Although communities require *polylingual workflows* (workflows that execute tools written in multiple, different computer languages) and *multiparty workflows*, adopting and managing different workflow systems is costly and difficult. In this work, we propose to tame this complexity through a common abstraction that covers most features used in practice and that is (or can be) implemented in many workflow systems.

In the computational workflow depicted in Figure 1, practitioners solved the problem by adopting the CWL standards. In this work, we posit that the CWL standards provide the common abstraction that can help overcome the main obstacles to sharing workflows between institutions and users. CWL achieves this by providing a declarative language that allows expressing computational workflows constructed from diverse software tools—each executed through their command-line interface, with the inputs and outputs of each tool clearly specified and with inputs possibly resulting from the execution of other tools. We also set out to introduce the CWL standards, with a threefold focus:

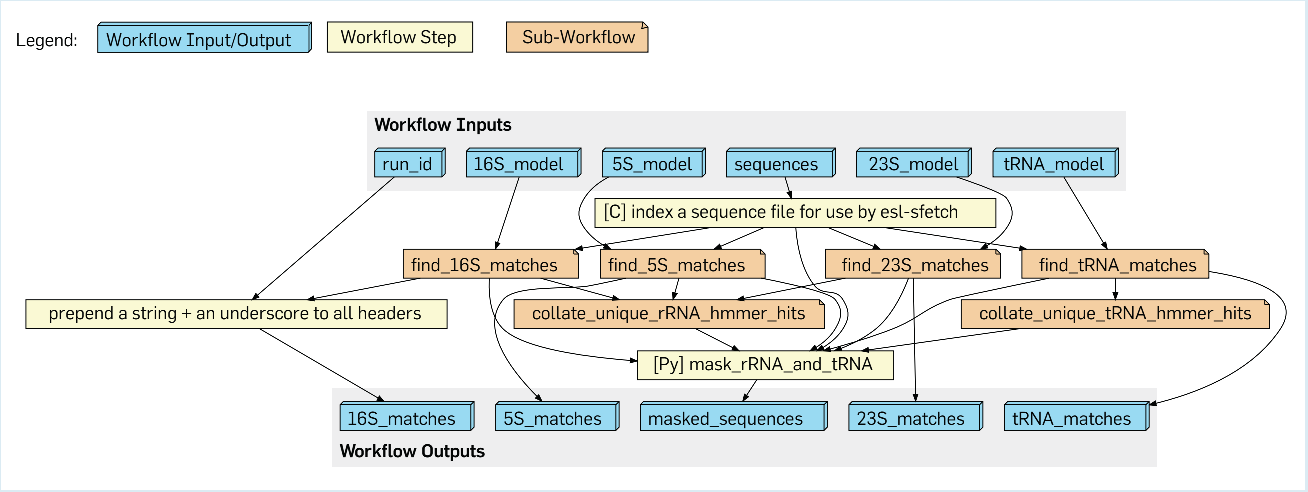
1. The CWL standards focus on maintaining a separation of concerns between the description and execution of tools and workflows, proposing a language that only includes operations commonly used across multiple communities of practice.
2. The CWL standards support workflow automation, scalability, abstraction, provenance, portability, and reusability.
3. To achieve these results, the CWL project takes a principled, community-first open source and open-standard approach.

The CWL standards are the product of an open and free standards-making community. While the CWL project began in bioinformatics, its many contributors shaped the standards to be useful in any domain that faces the problem of "many tools written in many programming languages by many parties." Since the ratification of the first version in 2016, the CWL standards have been used in other fields, including hydrology, radio astronomy, geo-spatial analysis,[13,23,32] and high-energy physics,[4] in addition to fast-growing bioinformatics fields such as metagenomics[27] and cancer research.[24] The CWL standards are featured in the IEEE 2791-2020 standard, sponsored and adopted by the U.S. FDA,[16] and the Netherlands' National Plan for Open Science.[34] A list of free and open source implementations of the CWL standards is offered in the Table. Multiple, commercially supported systems that follow the CWL standards for executing workflows are also available from vendors such as Curii (Arvados), DNAnexus, IBM (IBM® Spectrum LSF), Illumina (Illumina Connected Analytics), and Seven Bridges. The flexibility of the CWL standards enabled, for example, rapid collaboration on and prototyping of a COVID-19 public database and analysis resource.[15]

The separation of concerns proposed by the CWL standards enables diverse projects and can also benefit engineering and large industrial projects. Likewise, users of Docker or other software-container technologies that distribute analysis tools can leverage just the CWL Command Line Tool

Figure 1. Excerpt from a large microbiome bioinformatics CWL workflow.[27,*]

This part of the workflow, which is interpretable/executable on its own, aims to match the workflow inputs of genomic sequences to provided sequence models, which are dispatched to four sub-workflows (for instance, find_16S_matches). The sub-workflows are not detailed in the figure. Sub-workflow outputs are collated to identify unique sequence hits and then provided as overall workflow outputs. Arrows define the connection between tasks and imply their partial ordering, depicted here as layers of tasks that may execute concurrently. Workflow steps—for example, "mask_rRNA_and_tRNA"—execute command-line tools, shown here with indicators for their different programming languages ([Py] for Python, [C] for the C language).



* Diagram adapted from https://w3id.org/cwl/view/git/7bb76f33bf40b5cd2604001cac46f967a209c47f/workflows/rna-selector.cwl, which was originally retrieved from a corresponding CWL workflow of the EBI Metagenomics project, itself a conversion of the "rRNASelector"[25] program into a well-structured workflow, allowing for better parallelization of execution and provenance tracking.

standard to access a structured, workflow-independent description of how to run their tool(s) in the container, what data must be provided to the container, expected results, and where to find them.

Background on Workflows and Standards for Workflows

Workflows, and standards-based descriptions thereof, hold the potential to solve key problems in many domains of science and engineering.

Why workflows? In many domains, workflows include diverse analysis components, written in multiple, different computer languages by both end users and third parties. Such polylingual and multi-party workflows are already common or dominant in data-intensive fields, such as bioinformatics, image analysis, and radio astronomy. We envision they could bring important benefits to many other domains.

To thread data through analysis tools, domain experts such as bioinformaticians use specialized command-line interfaces,[12,31] while experts in other domains use proprietary, customized frameworks.[2,5] Workflow engines also help with efficiently managing the resources used to run scientific workloads.[7,10]

The workflow approach helps compose an entire application of these command-line analysis tools: Developers build graphical or textual descriptions of how to run these command-line tools, and scientists and engineers connect their inputs and outputs so that the data flows through. An example of a complex workflow problem is metagenomic analysis, for which Figure 1 illustrates a subset (a *sub-workflow*).

In practice, many research and engineering groups use workflows of the kind described in Figure 1. However, as highlighted in a "Technology Toolbox" article recently published in *Nature*,[29] these groups typically lack the ability to share and collaborate across institutions and infrastructures without costly manual translation of their workflows.

Using workflow techniques, especially with digital analysis processes, has become quite popular and does not appear to be slowing down. One workflow-management system, Galaxy Publication Library, recently celebrated its 10,000th citation, and more than 309 computational data-analysis workflow systems are known to exist.[b] A process, digital or otherwise, may grow to such complexity that its authors and users have difficulties understanding its structure, scaling and managing it, and keeping track of what happened in the past. Process dependencies may be undocumented, obfuscated, or otherwise effectively invisible. Outsiders or newcomers may find even an extensively documented process difficult to understand if it lacks a common framework or vocabulary. The need to run the process more frequently or with larger inputs is unlikely to be achieved by the initial entity—that is, either a script or a human—running the process. What seemed once a reasonable manual step—run this command here, paste the result there, and then call this person for permission—will become a bottleneck under the pressure of porting and reusing. Informal logs (if any) will quickly become unsuitable for helping an organization understand what happened, when, by whom, and to which data.

b Existing Workflow Systems: https://s.apache.org/existing-workflow-systems.

Monolingual and Polylingual Workflow Systems

Techniques for workflows can be implemented in many ways—that is, with varying degrees of formalism—which tends to correlate with execution flexibility and features. Whereas the most informal techniques typically require that all processing components are written in or are at least callable from the same programming language, formal workflow techniques tend to allow components to be developed in multiple programming languages.

Among the informal techniques, the do-it-yourself approach uses built-in capabilities from a particular programming language. For example, Python provides a *threading library*, and the Java-based Apache Hadoop[33] provides MapReduce capabilities. To gain flexibility when working with a particular programming language, general third-party libraries, such as *ipyparallel*,[a] can enable remote or distributed execution without having to rewrite one's code.

A more explicit workflow structure can be achieved by using a *workflow library* focusing on a specific programming language. For example, in Parsl,[2] the workflow constructs ("this is a unit of processing" or "here are the dependencies between the units") are made explicit and added by the developer to a Python script, to upgrade it to a scalable workflow. (While we list Parsl as an example of a monolingual workflow system, it also contains explicit support for executing external command-line tools.)

Two approaches—the use of per-language *add-in libraries* or the use of the *Portable Operating System Interface command-line interface (POSIX CLI)*[30]—can accommodate polylingual workflows, where components are written in more than one programming language or where components come from third parties and the user does not want to or cannot modify them. Using per-language add-in libraries entails either explicit function calls (for example, using Python *ctypes* to call a C library[b]) or the addition of annotations to the user's functions; this requires mapping/restricting to a common, cross-language data model.

Essentially all programming languages support the creation of POSIX CLIs, which are familiar to many Linux and macOS users as scripts or binaries that can be invoked on the shell with a set of arguments, reading and writing files, and executed in a separate process. Choosing the POSIX command-line interface as the coordination point means the connection between components is performed by an array of string arguments representing program options (including paths to data files) along with string-based environment variables (key-value pairs). Using the command line as a coordination interface has the advantage of not needing additional implementation in every programming language but is challenged by process start-up time and a very simple data model. (As a polylingual workflow standard, CWL uses the POSIX CLI data model.)

a IPython Parallel (ipyparallel) is a Python package and collection of CLI scripts for controlling clusters of IPython processes, built on the Jupyter protocol. See https://pypi.org/project/ipyparallel/.

b ctypes is a foreign function library for Python. See https://docs.python.org/3/library/ctypes.html.

Selected F/OSS workflow runners and platforms that implement the CWL standards.

Implementation	Platform Support
cwltool	Linux, macOS, MS Windows (via WSL 2) local execution only
Arvados	In the cloud on AWS, Azure, and Google Cloud Platform (GCP), on-premise and hybrid clusters using Slurm or LSF
Toil[35]	AWS, Azure, GCP, Grid Engine, HTCondor, IBM Spectrum LSF, Mesos, OpenStack, Slurm, PBS/Torque; also local execution on Linux, macOS, and MS Windows (via WSL 2)
CWL-Airflow[21]	Local execution on Linux, OS X, or via dedicated Airflow-enabled cluster.
StreamFlow[6]	Kubernetes, HPC with Singularity (PBS, Slurm), Occam, multi-node SSH, and local-only (Docker, Singularity)
REANA	Kubernetes

Workflow techniques aim to solve these problems by providing the abstraction, scaling, automation, and provenance (ASAP) features.[8] Workflow constructs enable a clear abstraction about the components, the relationships between them, and the inputs and outputs of the components turning them into well-labeled tools with documented expectations. This abstraction enables:

- **Scaling:** Execution can be parallelized and distributed.
- **Automation:** The abstraction can be used by a workflow engine to track, plan, and manage task execution.
- **Provenance tracking:** Descriptions of tasks, executors, inputs, and outputs—with timestamps, identifiers (unique names), and other logs—can be stored in relation to each other to later answer structured queries.

Why workflow standards? Although workflows are very popular, prior to the CWL standards, all workflow systems were incompatible with each other. This means that users who do not use the CWL standards are required to express their computational workflows in a different way each time they use another workflow system, leading to local success but global unportability.

The success of workflows is now their biggest drawback. Users are locked into a particular vendor, project, and often a specific hardware setup, hampering sharing and reuse. Even non-academics suffer from this situation, as the lack of standards, or their adoption, hinders effective collaboration on computational methods within and between companies. Likewise, this unportability affects public/private partnerships and the potential for technology transfer from public researchers.

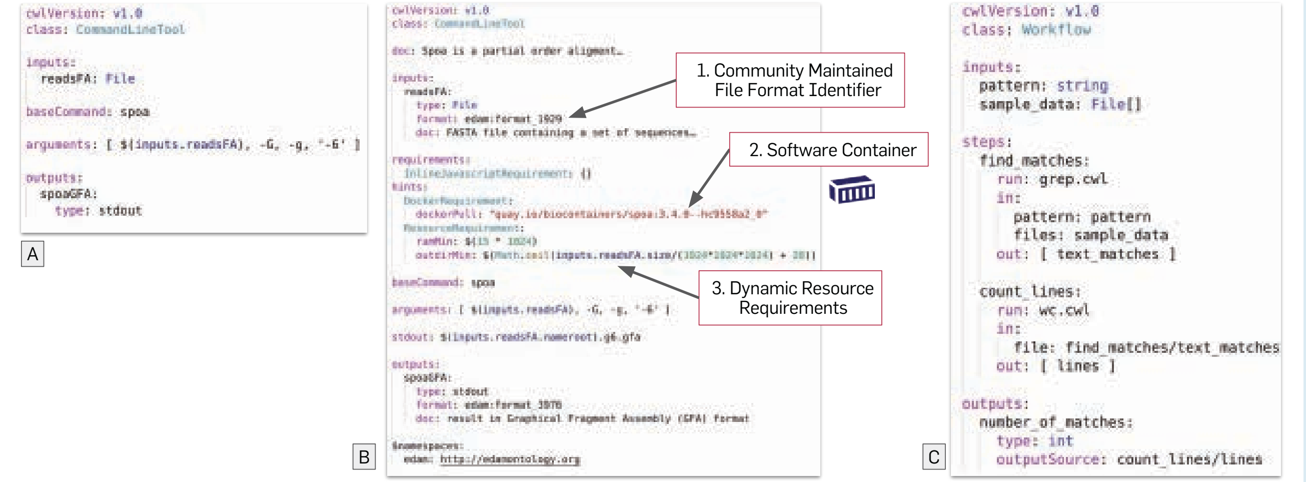
A second significant problem is that incomplete method descriptions are common when computational analysis is reported in academic research.[17] Reproduction, reuse, and replication[11] of these digital methods requires a complete description of which computer applications were used, how they were used, and how they were connected to each other. For precision and interoperability, this description should also be in an appropriate, standardized, machine-readable format.

A standard for sharing and reusing workflows can provide a solution to describing portable, reusable workflows while also being workflow-engine and vendor neutral.

Sharing workflow descriptions based on standards also addresses the second problem: The availability of the workflow description provides needed information when sharing, and the quality of the description provided by a structured, standards-based approach is much higher than the current approach of casual, unstructured, and almost always incomplete descriptions in scientific reports. Moreover, the operational parts of the

Figure 2. Example of CWL syntax and progressive enhancement.

(a) and (b) describe the same tool, but (b) is enhanced with additional features: human-readable documentation; file format identifiers for better validation of workflow connections; recommended software container image for more reproducible results and easier installation; and dynamically specified resource requirements to optimize task scheduling and resource usage without manual intervention. Resource requirements are expressed as *hints*. (c) shows an example of CWL Workflow syntax, where the underlying tool descriptions ("grep.cwl" and "wc.cwl") are in external files for ease of reuse.



description can be automated by the workflow-management system rather than by domain experts.

While (data) standards are commonly adopted and have become expected for funded projects in knowledge representation fields, the same cannot yet be said about workflows and workflow engines.

Features of the Common Workflow Language Standards

The Common Workflow Language standards aim to cover the common needs of users and the commonly implemented features of workflow runners or platforms. The remainder of this section presents an overview of CWL features, how they translate to executing workflows in CWL format, and where the CWL standards are not helpful.

The CWL standards support polylingual and multi-party workflows, for which they enable computational reuse and portability. To do so, each release of the CWL standards has two[c] main components: (1) a standard for describing command-line tools and (2) a standard for describing workflows that compose such tool descriptions. The goal of the *CWL Command Line Tool Description Standard* is to describe how a particular command-line tool works: What are the inputs and parameters and their types? How do you add the correct flags and switches to the command-line invocation? Where do you find the output files?

The CWL standards define an *explicit language*, both in syntax and in its data and execution model. Its textual syntax, derived from YAML,[d] does not restrict the amount of detail. For example, Figure 2a depicts a simple example with sparse detail, and Figure2b depicts the same example but with the execution augmented with more details. Each input to a tool has a name and a type—for instance, File (see Figure 2b, Item 1). Tool-description authors are encouraged to include documentation and labels for all components (as shown in Figure 2b), to enable the automatic generation of helpful visual depictions and even graphical user interfaces (GUIs) for any given CWL description. Metadata about the tool-description authors encourages attribution of their efforts. As shown in Figure 2b, Item 3, these tool descriptions can contain well-defined hints or mandatory requirements, such as which software container to use or the amount of required compute resources: memory, number of CPU cores, amount of disk space, and/or the maximum time or deadline to complete the step or entire workflow.

The CWL execution model is explicit. Each tool's runtime environment is explicit, and any required elements must be specified by the CWL tool-description author (in contrast to hints, which are optional).[e] Each tool invocation uses a separate working directory, populated according to the CWL tool description—for example, with the input files explicitly specified by the workflow author. Some applications expect particular filenames, directory layouts, and environment variables, and there are additional constructs in the CWL Command Line Tool standard to satisfy their needs.

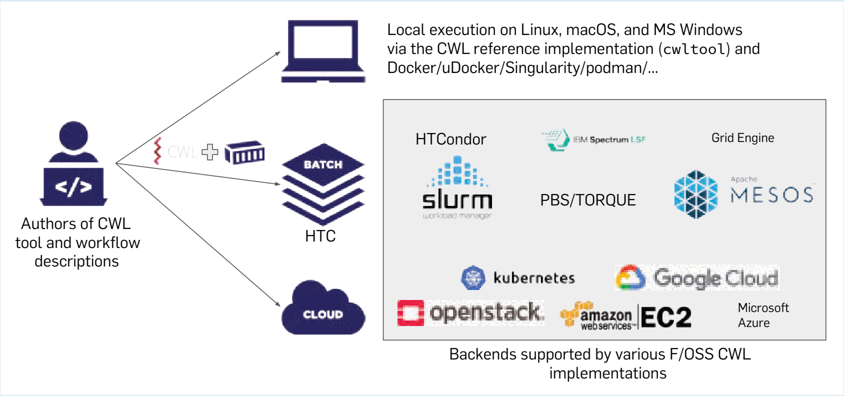
The explicit runtime model enables portability, by being explicit about data locations. As Figure 3 indicates, this enables the execution of CWL workflows in diverse environments as provided by various implementations of the CWL standards: the local environment of the author-scientist (for instance, a single desktop computer, laptop, or workstation), a remote batch production environment (for example, a cluster, an entire data center, or even a global multi-data center infrastructure), and an on-demand cloud environment.

The CWL standards explicitly support the use of *software container* technologies, such as Docker and Singularity, to enable the portability of the underlying analysis tools. Figure 2b, Item 2 illustrates the process of pulling a Docker container image from the Quay.io registry; then, the workflow engine automates the mounting of files and folders within the container. The container included in the figure has been developed by a trusted author and is commonly used in the bioinformatics field, with the expectation that its results are reproducible. Indeed, the use of containers can be seen as a confirmation that a tool's execution is reproducible when using only its explicitly declared runtime environment. Similarly, when *distributed execution* is desired, no changes to the CWL tool description are needed. File or directory inputs are already explicitly defined in the CWL description, so the (distributed) workflow runner can handle job placement and data routing between compute nodes without additional configuration.

Via these two features—special han-

Figure 3. Example of CWL portability.

The same workflow description runs on the scientist's own laptop or single machine, on any batch-production environment, and on any common public or private cloud. The CWL standards enable execution portability by being explicit about data locations and execution models.



c The third component, Schema Salad, is only of interest to those who want to parse the syntax of the schema language that is used to define the syntax of CWL itself.

d JSON is an acceptable subset of YAML, and common when converting from another format to CWL syntax.

e For details on how the CWL Command Line Tool standard specifies that tool executors should set up and control the runtime environment, visit: https://w3id.org/cwl/v1.2/CommandLineTool.html#Runtime_environment, which also specifies which directories tools are allowed to write to.

The CWL Project and Free/Open Source Software (F/OSS)

***Free and Open Source implementations of the CWL standards*. As of 2021, the CWL standards have gained much traction and are widely supported in practice. In addition to the implementations in the Table, Galaxy[1] and Pegasus[10] also have in-development support for the CWL standards.**

Wide adoption benefits from our principles: The CWL standards include conformance tests, but the CWL community does not yet test or certify implementations of the standards or specific technology stacks. Instead, the authors and service providers of workflow runners and workflow-management systems self-certify support for the CWL standards, based on a particular technology configuration they deploy and maintain.

***F/OSS tools and libraries for working with CWL-format documents*.[a] CWL plug-ins exist for Atom, Vim, Emacs, Visual Studio Code, IntelliJ, gedit, and any editor that supports the Language Server Protocol (LSP)[b] standard. There are tools to generate CWL syntax from Python (via argparse/click or via functions), ACD,[c] CTD,[d] and annotations in IPython Jupyter Notebooks. Libraries to generate and/or read CWL documents exist in many languages: Python, Java, R, Go, Scala, Javascript, Typescript, and C++.**

a Summarized from https://www.commonwl.org/tools/.

b https://microsoft.github.io/language-server-protocol/.

c "Ajax Command Definitions" as produced by the EMBOSS tools: http://emboss.sourceforge.net/developers/acd/.

d XML-based "Common Tool Descriptors"[9] originating in the OpenMS project: https://github.com/WorkflowConversion/CTDSchema.

dling of data paths and the optional but recommended use of software containers—the CWL standards enable portability (execution "without change"). While portability can be affected by various factors not controllable by software container technology—for instance, variation in the underlying operating-system kernel or in processor results—in practice, the exact same software container and data inputs lead to portability without further adjustment from the user.

To support features that are not in the CWL standards, the standards define *extension points* that permit namespaced, vendor-specific features in explicitly defined ways. If these extensions do not fundamentally change how the tool should operate, then they are added to the hints list, and other CWL-compatible engines can ignore them. However, if the extension is required to properly run the tool being described—for instance, due to the need for some specialized hardware—then the extension is listed under requirements, and CWL-compatible engines can recognize and explicitly declare their inability to execute that CWL description.

The CWL Workflow Description Standard builds upon the CWL Command Line Tool Standard. It has the same YAML- or JSON-style syntax, with explicit workflow-level inputs, outputs, and documentation (Figure 2c). Workflow descriptions consist of a list of steps, comprising CWL Command Line Tools or CWL sub-workflows, each re-exposing their tool's required inputs. Inputs for each step are connected by referencing the name of either the common *workflow inputs* or of outputs from other steps. The *workflow outputs* expose selected outputs from workflow steps, making explicit which intermediate-step outputs will be returned from the workflow. All connections include identifiers, which CWL document authors are encouraged to name meaningfully—for example, "reference_genome" instead of "input7."

CWL workflows form explicit dataflows, as required for a particular computational analysis. The connectivity between steps defines the partial execution order. Parallel execution of steps is permitted and encouraged whenever multiple steps have all their inputs satisfied. For example, in Figure 1, "find_16S_matches" and "find_S5_matches" are at the same data-dependency level and can execute concurrently or sequentially in any order. Additionally, a *scatter* construct allows the repeated execution of a CWL step (perhaps overlapping in time, depending on the available resources), where most of the inputs are the same except for one or more inputs that vary. This is done without having to modify the underlying tool description. Starting with CWL version 1.2, workflows can also conditionally skip execution of a step (tool or workflow), based upon a specified intermediate input or custom Boolean evaluation. Combining these features allows for a flexible *branch* mechanism, which allows workflow engines to calculate data dependencies before the workflow starts and thus retains the predictability of the dataflow paradigm.

In contrast to hard-coded approaches that rely on implicit file paths specific to each workflow, CWL workflows are more flexible, reusable, and portable, which enables scalability. The use of explicit runtime environments in the CWL standards, combined with explicit inputs/outputs to form the dataflow, enables step reordering and explicit handling of iterations. The same features enable scalable remote execution and, more generally, flexible use of runtime environments. Moreover, individual tool definitions from multiple workflows can be reused in any new workflow.

CWL workflow descriptions are also future proof. Forward compatibility of CWL documents is guaranteed, as each CWL document declares which version of the standards it was written for, and minor versions do not alter the required features of the major version. A standalone upgrader can automatically upgrade CWL documents from one version to the next, and many CWL-aware platforms will internally update user-submitted documents at runtime.

Execution of workflows in CWL format. CWL is a set of standards, not a particular software product to install, purchase, or rent. The CWL standards need to be implemented to be useful; a list of some implementations of the CWL standards is in the Table. Workflow/tool runners that claim compliance with the CWL standards are allowed significant flexibility in how and where they execute a user's CWL documents as long as they fulfill the requirements written in those documents. For example, they are allowed (and encouraged) to distribute execution of a workflow across all available computers that can fulfill user-specified resource requirements. Aspects of execution not defined by the CWL standards include Web APIs for workflow execution and real-time monitoring.

For example, details about when a step should be considered ready for execution are available in Section 4 of the CWL Workflow Description standard, but once all the inputs are available, the exact timing is up to the workflow engine itself.

Step execution may result in a temporary or permanent failure, as defined in Section 4 of the CWL Workflow Description standard. The workflow engine must control any automatic failure recovery attempts—for instance, to re-execute a workflow step. Most workflow engines that implement the CWL standards feature the ability to attempt several re-executions, set by the user, before reporting permanent failure.

The CWL community has developed the following optimizations without requiring that users rewrite their workflows to benefit:

- Automatic streaming of data inputs and outputs instead of waiting for all data to be downloaded or uploaded (where those data inputs or outputs are marked with "streamable: true").
- Workflow step placement based on data location,[18] resource needs, and/or cost of data transfer.[19]
- The reuse of the results from previously computed steps, even from a different workflow, as long as the inputs are identical. This can be controlled by the user via the "WorkReuse" directive in the CWL Workflow Standard.

Real-world usage at scale. CWL users and vendors routinely report that they analyze 5,000 whole-genome sequences in a single workflow execution. One customer of a commercial vendor reported a successful workflow run containing an 8,000-wide step; the entire workflow had 25,000 container executions. By design, the CWL standards do not impose any technical limitations on the size of files processed or to the number of tasks run in parallel. The major scalability bottlenecks are hardware-related—not having enough machines with enough memory, compute power, or disk space to process ever-growing data at a greater scale. As these boundaries move in the future with technological advances, the CWL standards should be able to keep up and not be a limitation.

When is CWL not useful? The CWL standards were designed for a particular style of command-line, tool-based data analysis. Therefore, the following situations are out of scope and not appropriate (or possible) to describe using CWL syntax:

- Safe interaction with stateful (web) services.
- Real-time communication between workflow steps.
- Interactions with command-line tools beside 1) constructing the command line and making available file inputs (both user-provided and synthesized from other inputs just prior to execution) and 2) consuming the output of the tool once its execution is finished, in the form of files created/changed, the POSIX standard output and error streams, and the POSIX exit code of the tool.
- Advanced control-flow techniques beyond conditional steps.
- Runtime workflow graph manipulations: dynamically adding or removing new steps during workflow execution, beyond any predefined conditional step execution tests that are in the original workflow description.
- Workflows that contain cycles: "Repeat this step or sub-workflow a specific number of times" or "Repeat this step or sub-workflow until a condition is met."[f]
- Workflows that need specific steps run on a specific day or at a specific time.

Open Source, Open Standards, Open Community

Given the numerous and diverse set of potential users, implementers, and other stakeholders, we posit that a project like CWL requires the combined development of code, standards, and community. Indeed, these requirements were part of the foundational design principles for CWL; in the long run, these principles have fostered free and open source software (see sidebar "The CWL Project and Free/Open Source Software") and a vibrant and active ecosystem.

The CWL principles. The CWL project is based on a set of five principles:

- **Principle 1:** At the core of the project is the community of people who care about its goals.
- **Principle 2:** To achieve the best possible results, there should be few, if any, barriers to participation. Specifically, to attract people with diverse experiences and perspectives, there must be no cost to participate.
- **Principle 3:** To enable the best outcomes, project outputs should be used as people see fit. Thus, the standards themselves must be licensed for reuse, with no acquisition price.
- **Principle 4:** The project must not favor any one company or group over another, but neither should it try to be all things to all people. The community decides.
- **Principle 5:** Concepts and ideas must be tested frequently. Tested and functional code is the beginning of evaluating a proposal, not the end.

Over time, CWL project members learned that this approach is a superset of the OpenStand Principles, a joint "Modern Paradigm for Standards" promoted by the IAB, IEEE, IETF, Internet Society, and W3C. The CWL project additions to the OpenStand Principles are 1) to keep participation free of cost, and 2) the explicit choice of Apache License 2.0 for all its text, conformance tests, and reference implementations.

Necessary and sufficient. All these principles have proven to be essential for the CWL project. For example, Principles 2 and 3 have enabled many implementations of the CWL standards, several of which reuse different parts of the reference implementation of the CWL standards (*reference runner*). Being community-first, per Principle 1, has led participants to create several projects that are outside the CWL standards; the most important contributions have made their way back into the project (Principle 4).

As part of Principle 5, contributors to the CWL project have developed a

f Supporting cycles/loops as an optional feature has been suggested for a future version of the CWL standards, but it has yet to be put forth as a formal proposal with a prototype implementation. As a work around, one can launch a CWL workflow from within a workflow system that does support cycles, as documented in the eWaterCycle case study with Cylc.[28]

suite of conformance tests for each version of the CWL standards. These publicly available tests were critical to the CWL project's success: They helped assess the reference implementation of the CWL standards, they provided early adopters with concrete examples, and they enabled developers and users of production implementations of the CWL standards to confirm their accuracy.

The CWL ecosystem. Beyond the ratified initial and updated CWL standards released over the last six years, the CWL community has developed many tools, software libraries, and connected specifications, and has shared CWL descriptions for popular tools. For example, there are software development kits (SDKs) for both Python and Java that are generated automatically from the CWL schema. This allows programmers to load, modify, and save CWL documents using an object-oriented model that directly corresponds to the standards themselves. CWL SDKs for other languages are possible by extending the code generation routines.[g] (See Sidebar: The CWL Project and Free/Open Source Software for practical details.)

The CWL standards offer strong support for the acute need to reuse (and, correspondingly, to share) information on workflow execution as well as on authoring and provenance. The *CWLPROV* prototype was created to show how existing standards[3,22,26] can be combined to represent the provenance of a specific execution of a CWL workflow.[20] Although, to date, CWLProv has only been implemented in the CWL reference runner, interest in further development and implementation is high.

Conclusion

The problem of standardizing computational reuse is only increasing in prominence and impact. Addressing this problem, various domains in science, engineering, and commerce have already started migrating to workflows, but efforts focusing on the portability and even definition of workflows remain scattered. In this work, we raise awareness to this problem and propose a community-driven solution.

By design, the CWL standards do not impose any technical limitations on the size of files processed or to the number of tasks run in parallel.

The CWL is a family of standards for the description of command-line tools and of the workflows made from these tools. It includes many features developed in collaboration with the community: support for software containers, resource requirements, workflow-level conditional branching, and more. Built on a foundation of five guiding principles, the CWL project delivers open standards, open source code, and an open community.

For the past six years, the CWL community has grown organically. Organizations looking to write, use, or fund data-analysis workflows based upon command-line tools should adopt or even require the CWL standards, because they offer a common yet reduced set of capabilities that are both used in practice and implemented in many popular workflow systems. There are other ways CWL offers value: It is supported by a large-scale community, diverse fields have already adopted it, and its adoption is rapidly growing. Specifically:

1. With a reduced set of capabilities, the CWL standards limit the complexity encountered by new users when they first start and by operators during implementation. Feedback from the community indicates these are appreciated.

2. CWL's use of declarative syntax allows users to specify workflows even if they do not know exactly where the workflows would (later) run.

3. The CWL project is governed in the public interest and produces freely available open standards. The CWL project itself is not a specific workflow-management system, workflow runner, or vendor. This allows potential users, operators, and vendors to avoid lock-in and be more flexible in the future.

4. By offering standards, the CWL project distinguishes itself, especially for the complex interactions that appear in scientific and engineering collaborations. These interactions include defining workflows from many different tools (or steps), sharing workflows, long-term archiving, fulfilling requirements of regulators (for example, U.S. FDA), and making workflow executions auditable and reproducible. This is especially useful in cooperative environments, where groups that compete

g See the *codegen*.py files in https://pypi.org/project/schema-salad/7.1.20210316164414/.

also need to collaborate, or in scientific papers where results can be reused very efficiently if the analysis is described in a CWL workflow with publicly available software containers for all steps.

5. The CWL standards are already implemented, adopted, and used, with many production-grade implementations available as open source and with zero-cost. Thus, the different communities of users of the CWL standards already offer numerous workflow and tool descriptions. This is akin to how the Python ecosystem of shared libraries, code, and recipes is already helpful.

This is a call for others to embrace workflow thinking and join the CWL community in creating and sharing portable and complete workflow descriptions. With the CWL standards, the methods are included and ready to (re)use.

Acknowledgments

The list of those involved in the CWL project we would like to acknowledge is extensive. A comprehensive list is available in the online supplementary material: https://bit.ly/3MoOPfQ. **Funding acknowledgments:** European Commission grants BioExcel-2 (SSR) H2020-INFRAEDI-02-2018 823830, BioExcel (SSR) H2020-EINFRA-2015-1 675728, EOSC-Life (SSR) H2020-INFRAEOSC-2018-2 824087, EOSCPilot (MRC) H2020-INFRADEV-2016-2 739563, IBISBA 1.0 (SSR) H2020-INFRAIA-2017-1-two-stage 730976, ELIXIR-EXCELERATE (SSR, HM) H2020-INFRADEV-1-2015-1 676559, ASTERICS (MRC) INFRADEV-4-2014-2015. ELIXIR the research infrastructure for life-science data, Interoperability Platform Implementation Study (MRC). 2018-CWL. Various universities have also co-sponsored this project. We thank Vrije Universiteit of Amsterdam, the Netherlands, where the first three authors have their primary affiliation.

References

1. Afgan, E. et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Research 46*, W1 (July 2018), W537–W544. https://doi.org/10.1093/nar/gky379.
2. Babuji, Y. et al. Parsl: Pervasive parallel programming in Python. In *Proceedings of the 28th Intern. Symp. on High-Performance Parallel and Distributed Computing*. Association for Computing Machinery (2019), 25–36. https://doi.org/10.1145/3307681.3325400.
3. Belhajjame, K. et al. Using a suite of ontologies for preserving workflow-centric research objects. *J. of Web Semantics 32* (May 2015), 16–42. https://doi.org/10.1016/j.websem.2015.01.003.
4. Bell, T. et al. *Web-based Analysis Services Report.* Technical Report CERN-IT-Note-2018-004. (2017), CERN, Geneva, Switzerland. http://cds.cern.ch/record/2315331/.
5. Berthold, M.R et al. KNIME–The Konstanz information miner: Version 2.0 and beyond. *ACM SIGKDD Explorations Newsletter 11*, 1 (Nov. 2009), 26–31. https://doi.org/10.1145/1656274.1656280.
6. Colonnelli, I. et al. StreamFlow: Cross-breeding cloud with HPC. *IEEE Transactions on Emerging Topics in Computing* (2020), 1–1. https://doi.org/10.1109/TETC.2020.3019202.
7. Couvares, P. et al. Workflow management in Condor. In *Workflows for e-Science: Scientific Workflows for Grids*, I.J. Taylor, E. Deelman, D.B. Gannon, and M. Shields (Eds.). Springer, London (2007), 357–375. https://doi.org/10.1007/978-1-84628-757-2_22.
8. Cuevas-Vicenttín, C. et al. Scientific workflows and provenance: Introduction and research opportunities. *Datenbank-Spektrum 12*, 3 (Nov. 2012), 193–203. https://doi.org/10.1007/s13222-012-0100-z.
9. de la Garza, L. et al. From the desktop to the grid: Scalable bioinformatics via workflow conversion. *BMC Bioinformatics 17*, 1 (March 2016), 127. https://doi.org/10.1186/s12859-016-0978-9.
10. Deelman, E. et al. Pegasus, a workflow management system for science automation. *Future Generation Computer Systems 46* (May 2015), 17–35. https://doi.org/10.1016/j.future.2014.10.008
11. Feitelson, D.G. From repeatability to reproducibility and corroboration. *ACM SIGOPS Operating Systems Review 49*, 1 (Jan. 2015), 3–11. https://doi.org/10.1145/2723872.2723875.
12. Georgeson, P. et al. Bionitio: Demonstrating and facilitating best practices for bioinformatics command-line software. *GigaScience 8*, giz109 (Sept. 2019). https://doi.org/10.1093/gigascience/giz109.
13. Gonçalves, P. OGC Earth observations applications pilot: Terradue engineering report. OGC Public Engineering Report OGC 20-042. Open Geospatial Consortium. http://docs.opengeospatial.org/per/20-042.html.
14. Gryk, M.R. and Ludäscher, B. Workflows and provenance: Toward information science solutions for the natural sciences. *Library Trends 65*, 4 (2017), 555–562. https://doi.org/10.1353/lib.2017.0018.
15. Guarracino, A. et al. COVID-19 PubSeq: Public SARS-CoV-2 sequence resource. Bioinformatics Open Source Conference (July 2020), https://sched.co/coLw.
16. IEEE standard for bioinformatics analyses generated by high-throughput sequencing (HTS) to facilitate communication. (May 11, 2020), https://doi.org/10.1109/IEEESTD.2020.9094416.
17. Ivie, P. and Thain, D. Reproducibility in scientific computing. *ACM Computing Surveys 51*, 3 (July 2018), 63:1–63:36. https://doi.org/10.1145/3186266.
18. Jiang, F. Castillo, C., and Ahalt, S. TR-19-01: A cloud-agnostic framework for geo-distributed data-intensive applications. RENCI, University of North Carolina at Chapel Hill, (2019), https://renci.org/technical-reports/tr-19-01/.
19. Jiang, F., Ferriter, K., and Castillo, C. PIVOT: Cost-aware scheduling of data-intensive applications in a cloud-agnostic system. RENCI, University of North Carolina at Chapel Hill, (2019). https://renci.org/technical-reports/tr-19-02/.
20. Khan, F.Z. et al. Sharing interoperable workflow provenance: A review of best practices and their practical application in CWLProv. *GigaScience 8*, 11 (November 2019), giz095. https://doi.org/10.1093/gigascience/giz095.
21. Kotliar, M., Kartashov, A.V., and Barski, A. CWL-Airflow: A lightweight pipeline manager supporting Common Workflow Language. *GigaScience 8*, 7 (July 2019), giz095. https://doi.org/10.1093/gigascience/giz084.
22. Kunze, J., Littman, J., Madden, E., Scancella, J., and Adams, C. The BagIt file packaging format (V1.0). (October 2018), DOI 10.17487/RFC8493. https://www.rfc-editor.org/info/rfc8493.
23. Landry, T. OGC Earth observation applications pilot: CRIM engineering report. Open Geospatial Consortium Public Engineering Report 20-045 (2020), http://docs.opengeospatial.org/per/20-045.html
24. Lau, J.W. et al. The Cancer Genomics Cloud: Collaborative, reproducible, and democratized—A new paradigm in large-scale computational research. *Cancer Research 77*, 21 (Oct. 2017), e3–e6. https://doi.org/10.1158/0008-5472.can-17-0387.
25. Lee, J-H., Yi, H., and Chun, J. rRNASelector: A computer program for selecting ribosomal RNA encoding sequences from metagenomic and metatranscriptomic shotgun libraries. *The J. of Microbiology 49*, 4 (September 2011), 689. https://doi.org/10.1007/s12275-011-1213-z.
26. Missier, P., Belhajjame, K., and Cheney, J. The W3C PROV family of specifications for modelling provenance metadata. In *Proceedings of the 16th Intern. Conf. on Extending Database Technology*. Association for Computing Machinery (2013). https://doi.org/10.1145/2452376.2452478.
27. Mitchell, A-L. MGnify: The microbiome analysis resource in 2020. *Nucleic Acids Research 48*, D1 (January 2020), D570–D578. https://doi.org/10.1093/nar/gkz1035.
28. Oliver, H. Workflow automation for cycling systems: The Cylc Workflow Engine. *Computing in Science Engineering* (2019), 1–1. https://doi.org/10.1109/MCSE.2019.2906593 00000.
29. Perkel, J.M. Workflow systems turn raw data into scientific knowledge. *Nature* 573 (September 2019), 149–150. https://doi.org/10.1038/d41586-019-02619-z.
30. POSIX.1-2008: IEEE Std 1003.1™-2008 and The Open Group Technical Standard Base Specifications, Issue 7. IEEE and The Open Group, https://pubs.opengroup.org/onlinepubs/9699919799.2008edition/.
31. Seemann, T. Ten recommendations for creating usable bioinformatics command line software. *GigaScience* 2, 2047-217X-2-15 (December 2013). https://doi.org/10.1186/2047-217X-2-15.
32. Simonis, I. OGC Earth observation applications pilot: Summary engineering report. Open Geospatial Consortium Public Engineering Report OGC 20-073 (2020), https://docs.ogc.org/per/20-073.html.
33. Taylor, R.C. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. *BMC Bioinformatics 11*, 12 (December 2010), S1. https://doi.org/10.1186/1471-2105-11-S12-S1.
34. van Wezenbeek, W.J.S.M., Touwen, H.J.J., Versteeg, A.M.C., and van Wesenbeeck, A.J.M. National Open Science Plan. Ministry of Education, Culture, and Science, Netherlands, (2017). https://doi.org/10.4233/uuid:9e9fa82e-06c1-4d0d-9e20-5620259a6c65.
35. Vivian, J. Toil enables reproducible, open source, big biomedical data analyses. *Nature Biotechnology 35*, 4 (April 2017), 314–316. https://doi.org/10.1038/nbt.3772.

more online

For additional information, access the supplementary material for this article at https://dl.acm.org/doi/10.1145/3486897.

Michael R. Crusoe is a promovendus at VU Amsterdam, Department of Computer Science, Netherlands.; CWL Project Lead at Software Freedom Conservancy, Inc., USA; and Project Leader Compute Platform in ELIXIR-NL at DTL Projects, Utrecht, Netherlands.

Sanne Abeln is an associate professor in Bioinformatics at VU Amsterdam, Department of Computer Science, Netherlands.

Alexandru Iosup is a university research chair and full professor at VU Amsterdam, Department of Computer Science, Netherlands.

Peter Amstutz is a principal software engineer at Curii Corporation, Sommerville, MA, USA.

John Chilton is a computational scientist at the Nekrutenko Lab at Pennsylvania State University Department of Biochemistry and Molecular Biology, State College, PA, USA.

Nebojša Tijanić was a software engineer at Seven Bridges Genomics Inc., Charlestown, MA, USA.

Hervé Ménager is a research engineer at the Institut Pasteur, Université de Paris, Bioinformatics and Biostatistics Hub, F-75015, Paris, France.

Stian Soiland-Reyes is a technical architect at The University of Manchester, Department of Computer Science, Manchester, U.K.; and a Ph.D. candidate at the Informatics Institute, University of Amsterdam, Netherlands.

Bogdan Gavrilović is a product development director at Seven Bridges Genomics Inc., Charlestown, MA, USA.

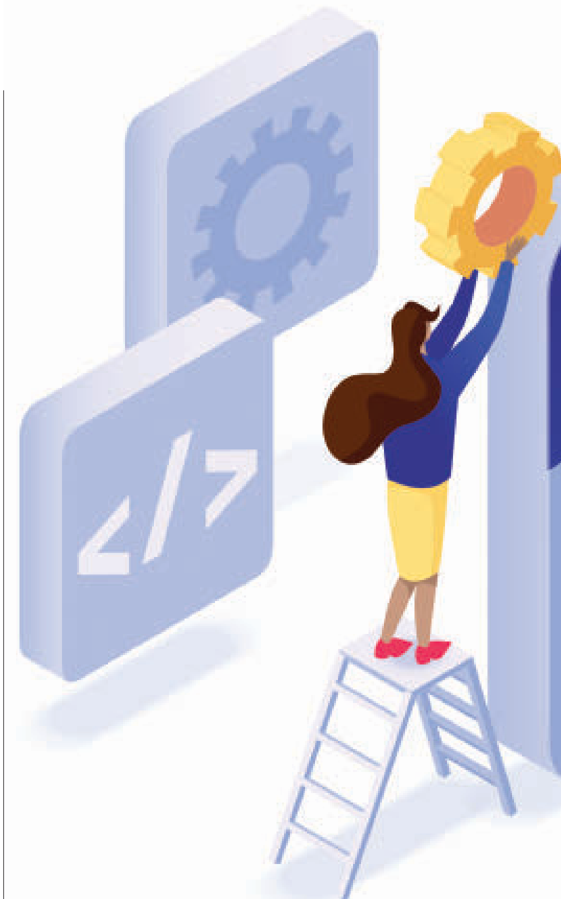
Carole Goble (CBE FREng FBCS CITP) is a full professor of Computer Science at The University of Manchester, Department of Computer Science, Manchester, U.K.

DOI:10.1145/3488717

Perspectives on the role and responsibility of the data-management research community in designing, developing, using, and overseeing automated decision systems.

BY JULIA STOYANOVICH, SERGE ABITEBOUL, BILL HOWE, H.V. JAGADISH, AND SEBASTIAN SCHELTER

Responsible Data Management



INCORPORATING ETHICS AND legal compliance into data-driven algorithmic systems has been attracting significant attention from the computing research community, most notably under the umbrella of fair[8] and interpretable[16] machine learning. While important, much of this work has been limited in scope to the "last mile" of data analysis and has disregarded both the *system's design, development, and use life cycle* (What are we automating and why? Is the system working as intended? Are there any unforeseen consequences post-deployment?) and the *data life cycle* (Where did the data come from? How long is it valid and appropriate?). In this article, we argue two points. First, the decisions we make during data collection and preparation profoundly impact the robustness, fairness, and interpretability of the systems we build. Second, our responsibility for the operation of these systems does not stop when they are deployed.

Example: Automated hiring systems. To make our discussion concrete, consider the use of predictive analytics in hiring. Automated hiring systems are seeing ever broader use and are as varied as the hiring practices themselves, ranging from resume screeners that claim to identify promising applicants[a] to video and voice analysis tools that facilitate the interview process[b] and game-based assessments that promise to surface personality traits indicative of future success.[c] Bogen and Rieke[5] describe the hiring process from the employer's point of view as a series of decisions that forms a funnel, with stages corresponding to

a https://www.crystalknows.com
b https://www.hirevue.com
c https://www.pymetrics.ai

sourcing, screening, interviewing, and selection. (Figure 1 depicts a slightly reinterpreted version of that funnel.)

The popularity of automated hiring systems is due in no small part to our collective quest for efficiency. In 2019 alone, the global market for artificial intelligence (AI) in recruitment was valued at $580 million.[d] Employers choose to use these systems to source and screen candidates faster, with less paperwork, and, in the post-COVID-19 world, as little in-person contact as is practical. Candidates are promised a more streamlined job-search experience, although they rarely have a say in whether they are screened by a machine.

The flip side of efficiency afforded by automation is that we rarely understand how these systems work and, indeed, whether they work. Is a resumé screener identifying promising candidates or is it picking up irrelevant—or even discriminatory—patterns from historical data, limiting access to essential economic opportunity for entire segments of the population and potentially exposing an employer to legal liability? Is a job seeker participating in a fair competition if she is being systematically screened out, with no opportunity for human intervention and recourse, despite being well-qualified for the job?

If current adoption trends are any indication, automated hiring systems are poised to impact each one of us—as employees, employers, or both. What's

d https://www.industryarc.com/Report/19231/artificial-intelligence-in-recruitmentmarket.html

IMAGE BY PAVLO PLAKHOTIA

» key insights

- **Responsible data management involves incorporating ethical and legal considerations across the life cycle of data collection, analysis, and use in all data-intensive systems, whether they involve machine learning and AI or not.**
- **Decisions during data collection and preparation profoundly impact the robustness, fairness, and interpretability of data-intensive systems. We must consider these earlier life cycle stages to improve data quality, control for bias, and allow humans to oversee the operation of these systems.**
- **Data alone is insufficient to distinguish between a distorted reflection of a perfect world, a perfect reflection of a distorted world, or a combination of both. The assumed or externally verified nature of the distortions must be explicitly stated to allow us to decide whether and how to mitigate their effects.**

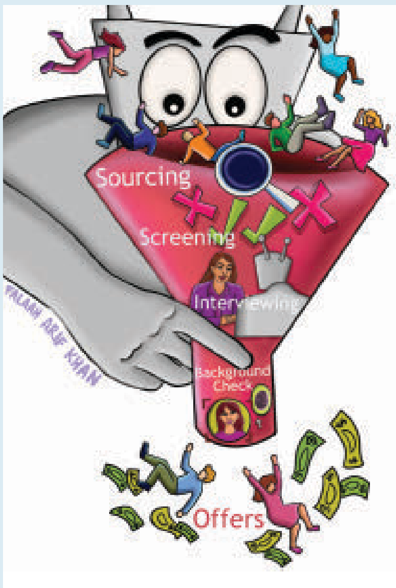


Figure 1. The hiring funnel is an example of an automated decision system—a data-driven, algorithm-assisted process that culminates in job offers to some candidates and rejections to others.

more, many of us will be asked to help design and build such systems. Yet, their widespread use far outpaces our collective ability to understand, verify, and oversee them. This is emblematic of a broader problem: the widespread and often rushed adoption of *automated decision systems* (ADSs) without an appropriate prior evaluation of their effectiveness, legal compliance, and social sustainability.

Defining ADSs. There is currently no consensus as to what an ADS is or is not, though proposed regulation in the European Union (EU), several U.S. states, and other jurisdictions are beginning to converge on some factors to consider: the degree of human discretion in the decision, the level of impact, and the specific technologies involved. As an example of the challenges, Chapter 6 of the New York City ADS Task Force report[e] summarizes a months-long struggle to, somewhat ironically, define its own mandate: to craft a definition that captures the breadth of ethical and legal concerns, yet remains practically useful. Our view is to lean towards breadth, but to tailor operational requirements and oversight mechanisms for an ADS depending on application domain and context of use, level of impact,[34] and relevant legal and regulatory requirements. For example, the use of ADSs in hiring and employment is subject to different concerns than their use in credit and lending. Further, the potential harms will be different depending on whether an ADS is used to advertise employment or financial opportunities or to help make decisions about whom to hire and to whom a loan should be offered.

e https://www1.nyc.gov/site/adstaskforce/index.page

To define ADS, we may start with some examples. Figure 1's hiring funnel and associated components, such as an automated resume screening tool and a tool that matches job applicants with positions, are natural examples of ADSs. But is a calculator an ADS? No, because it is not qualified with a context of use. Armed with these examples, we propose a pragmatic definition of ADSs:

- They process data about people, some of which may be sensitive or proprietary
- They help make decisions that are consequential to people's lives and livelihoods
- They involve a combination of human and automated decision-making
- They are designed to improve efficiency and, where applicable, promote equitable access to opportunity

In this definition, we deliberately direct our attention toward systems in which the ultimate decision-making responsibility is with a human and away from fully autonomous systems, such as self-driving cars. Advertising systems are ADSs; while they may operate autonomously, the conditions of their operation are specified and reviewed via negotiations between platform providers and advertisers. Further, regulation is compelling ever closer human oversight and involvement in the operations of such systems. Actuarial models, music recommendation systems, and health screening tools are all ADSs as well.

Why responsible data management? The placement of technical components that assist in decision-making—a spreadsheet formula, a matchmaking algorithm, or predictive analytics—within the *life cycle of data collection and analysis* is central to defining an ADS. This, in turn, uniquely positions the data-management community to deliver true practical impact in the responsible design, development, use, and oversight of these systems. Because data-management technology offers a natural, centralized point for enforcing policies, we can develop methodologies to enforce requirements transparently and explicitly through the life cycle of an ADS. Due to the unique blend of theory and systems in our methodological toolkit, we can help inform regulation by studying the feasible tradeoffs between different classes of legal and efficiency requirements. Our pragmatic approach enables us to support compliance by developing standards for effective and efficient auditing and disclosure, and by developing protocols for embedding these standards in systems.

In this article, we assert that the data-management community should play a central role in responsible ADS design, development, use, and oversight. Automated decision systems may or may not use AI, and they may or may not operate with a high degree of autonomy, but they all rely heavily on data. To set the stage for our discussion, we begin by interpreting the term "bias" (Section 2). We then discuss the data management-related challenges of ADS oversight and embedding responsibility into ADS life cycle management, pointing out specific opportunities for novel research contributions. Our focus is on specific issues where there is both a well-articulated need and strong evidence that technical interventions are possible. Fully addressing all the issues we raise requires socio-technical solutions that go beyond the scope of what we can do with technology alone. Although vital, since our focus is on technical data-management interventions, we do not discuss such socio-technical solutions in this article.

Crucially, the data-management problems we seek to address are not purely technical. Rather, they are socio-legal-technical. It is naïve to expect that purely technical solutions will suffice, so we must step outside our engineering comfort zone and start reasoning in terms of values and beliefs, in addition to checking results against known ground truths and optimizing for efficiency objectives. This seems high-risk, but one of the upsides is being able to explain to our children what we do and why it matters.

All About That Bias

We often hear that an ADS, such as an automated hiring system, operates on "biased data" and results in "biased outcomes." What is the meaning of the term "bias" in this context, how does it exhibit itself through the ADS life cycle, and what does data-management technology have to offer to help mitigate it?

Bias in a general sense refers to systematic and unfair discrimination against certain individuals or groups of individuals in favor of others. In their seminal 1996 paper, Friedman and Nissenbaum identified three types of bias that can arise in computer systems: *preexisting*, *technical*, and *emergent*.[12] We discuss each of these in turn in the remainder of this section, while also drawing on a recent fine-grained taxonomy of bias, with insightful examples that concern social media platforms, from Olteanu et al.[26]

Preexisting bias. This type of bias has its origins in society. In data-science applications, it exhibits itself in the input data. Detecting and mitigating preexisting bias is the subject of much research under the heading of algorithmic fairness.[8] Importantly, the presence or absence of this type of bias cannot be scientifically verified; rather, it must be postulated based on a belief system.[11] Consequently, the effectiveness—or even the validity—of a technical attempt to mitigate preexisting bias is predicated on that belief system. To explain preexisting bias and the limits of technical interventions, such as data debiasing, we find it helpful to use the mirror reflection metaphor, depicted in Figure 2.

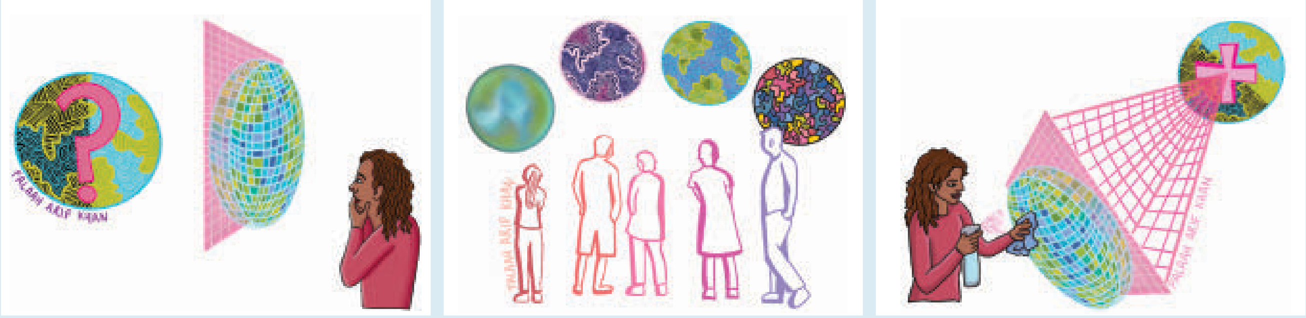
The mirror metaphor. Data is a mirror reflection of the world. When we think about preexisting bias in the data, we interrogate this reflection, which is often distorted. One possible reason is that the mirror (the measurement process) introduces distortions. It faithfully represents some portions of the world, while amplifying or diminishing others. Another possibility is that even a perfect mirror can only reflect a distorted world—a world such as it is, and not as it could or should be.

The mirror metaphor helps us make several simple but important observations. First, based on the reflection alone, and without knowledge about the properties of the mirror and of the world it reflects, we cannot know whether the reflection is distorted, and, if so, for what reason. That is, data alone cannot tell us whether it is a distorted reflection of a perfect world, a perfect reflection of a distorted world, or whether these distortions compound. The assumed or externally verified nature of the distortions must be explicitly stated, to allow us to decide whether and how to mitigate their effects. Our second observation is that it is up to people—individuals, groups, and society at large—and not data or algorithms, to come to a consensus about whether the world is how it should be or if it needs to be improved and, if so, how we should go about improving it. The third and final observation is that, if data is used to make important decisions, such as who to hire and what salary to offer, then compensating for distortions is worthwhile. But the mirror metaphor only takes us so far. We must work much harder—usually going far beyond technological solutions—to propagate the changes back into the world and not merely brush up the reflection.[37]

As an example of preexisting bias in hiring, consider the use of an applicant's Scholastic Assessment Test (SAT) score during the screening stage. It has been documented that the mean score of the math section of the SAT, as well as the shape of the score distribution, differs across racial groups.[28] If we believed that standardized test scores were sufficiently impacted by preparation courses and that the score itself says more about socioeconomic conditions than an individual's academic potential, then we would consider the data to be biased. We may then seek to correct for that bias before using the feature, for example, by selecting the top-performing individuals of each racial group, or by using a more sophisticated *fair ranking method* in accordance with our beliefs about the nature of the bias and with our bias mitigation goals.[40] Alternatively, we may disregard this feature altogether.

Technical bias. This type of bias arises due to the operation of the technical system itself, and it can amplify

Figure 2. Data as a mirror reflection of the world,[37] illustrated by Falaah Arif Khan.



(a) Data is an image of the world, its mirror reflection. When we think about bias in the data, we interrogate this reflection. Does the data systematically over-represent or under-represent some parts of the world, or does it distort reality in some other way?

(b) Even if we were able to reflect the world perfectly in the data, it would still be a reflection of the world as it is, not how it could or should be. People—not data or algorithms—must decide what world we want to live in.

(c) Debiasing data may be worthwhile if we are about to base consequential decisions on that data. But we must be careful to propagate the changes back into the world, not merely touch up its reflection.

preexisting bias. Technical bias, particularly when it is due to preprocessing decisions or post-deployment issues in data-intensive pipelines, has been noted as problematic,[23,26,33] but it has so far received limited attention when it comes to diagnostics and mitigation techniques. We now give examples of potential sources of technical bias in several ADS life cycle stages, which are particularly relevant to data management.

Data cleansing. Methods for missing-value imputation that are based on incorrect assumptions about whether data is missing at random may distort protected group proportions. Consider a form that gives job applicants a binary gender choice but also allows gender to be unspecified. Suppose that about half of the applicants identify as men and half as women, but that women are more likely to omit gender. If mode imputation—replacing a missing value with the most frequent value for the feature, a common setting in scikit-learn—is applied, then all (predominantly female) unspecified gender values will be set to male. More generally, multiclass classification for missing-value imputation typically only uses the most frequent classes as target variables,[4] leading to a distortion for small groups, because membership in these groups will not be imputed.

Next, suppose that some individuals identify as non-binary. Because the system only supports male, female, and unspecified as options, these individuals will leave gender unspecified. If mode imputation is used, then their gender will be set to male. A more sophisticated imputation method will still use values from the active domain of the feature, setting the missing values of gender to either male or female. This example illustrates that bias can arise from an incomplete or incorrect choice of data representation. While dealing with null values is known to be difficult and is already considered among the issues in data cleansing, the needs of responsible data management introduce new problems. It has been documented that data-quality issues often disproportionately affect members of historically disadvantaged groups,[20] so we risk compounding technical bias due to data representation with bias due to statistical concerns.

The flip side of efficiency afforded by automation is that we rarely understand how these systems work and, indeed, whether they work.

Other data transformations that can introduce skew include text normalization, such as lowercasing, spell corrections, or stemming. These operations can be seen as a form of aggregation, in effect collapsing terms with different meanings under the same representation. For example, lowercasing "Iris," a person's name, as "iris" will make it indistinguishable from the name of a flower or from the membrane behind the cornea of the eye, while stemming the terms "[tree] leaves" and "[he is] leaving" will represent both as "leav."[26]

Other examples of aggregation that can lead to data distribution changes include "zooming out" spatially or temporally: replacing an attribute value with a coarser geographic or temporal designation or mapping a location to the center of the corresponding geographical bounding box.[26]

Filtering. Selections and joins are commonly used as part of data preprocessing. A selection operation checks each data record against a predicate—for instance, U.S. address ZIP code is 10065 or age is less than 30—and retains only those records that match the predicate. A join combines data from multiple tables—for example, creating a record that contains a patient's demographics and clinical records using the social security number attribute contained in both data sources as the join key. These operations can arbitrarily change the proportion of protected groups (for example, female gender) even if they do not directly use the sensitive attribute (for example, gender) as part of the predicate or the join key. For example, selecting individuals whose mailing address ZIP code is 10065—one of the most affluent locations on Manhattan's Upper East Side—may change the data distribution by race. Similarly, joining patient demographic data with clinical records may introduce skew by age, with fewer young individuals having matching clinical records. These changes in proportion may be unintended but are important to detect, particularly when they occur during one of many preprocessing steps in the ADS pipeline.

Another potential source of technical bias is the use of pretrained word embeddings. For example, a pipeline may replace a textual name feature with the corresponding vector from a word embedding that is missing for rare, non-Western names. If we then filter out records for which no embedding was found, we may disproportionately remove individuals from specific ethnic groups.

Ranking. Technical bias can arise when results are presented in ranked order, such as when a hiring manager is considering potential candidates to invite for in-person interviews. The main reason is inherent position bias—the geometric drop in visibility for items at lower ranks compared to those at higher ranks—which arises because in Western cultures we read from top to bottom and from left to right: Items in the top-left corner of the screen attract more attention.[3] A practical implication is that, even if two candidates are equally suitable for the job, only one of them can be placed above the other, which implies prioritization. Depending on the application's needs and on the decision-maker's level of technical sophistication, this problem can be addressed by suitably randomizing the ranking, showing results with ties, or plotting the score distribution.

Emergent bias. This type of bias arises in the context of use of the technical system. In Web ranking and recommendation in e-commerce, a prominent example is "rich-get-richer": searchers tend to trust systems to show them the most suitable items at the top positions, which in turn shapes a searcher's idea of a satisfactory answer.

This example immediately translates to hiring and employment. If hiring managers trust recommendations from an ADS, and if these recommendations systematically prioritize applicants of a particular demographic profile, then a feedback loop will be created, further diminishing workforce diversity over time. Bogen and Rieke[5] illustrate this problem: "For example, an employer, with the help of a third-party vendor, might select a group of employees who meet some definition of success—for instance, those who 'outperformed' their peers on the job. If the employer's performance evaluations were themselves biased, favoring men, then the resulting model might predict that men are more likely to be high performers than women, or make more errors when evaluating women."

Emergent bias is particularly difficult to detect and mitigate, because it refers to the impacts of an ADS outside the systems' direct control. We will cover this in the "Overseeing ADS" section.

Managing the ADS Data Life Cycle

Automated decision systems critically depend on data and should be seen through the lens of the *data life cycle*.[19] Responsibility concerns, and important decision points, arise in data sharing, annotation, acquisition, curation, cleansing, and integration. Consequently, substantial opportunities for improving data quality and representativeness, controlling for bias, and allowing humans to oversee the process are missed if we do not consider these earlier life cycle stages.

Database systems centralize correctness constraints to simplify application development with the help of schemas, standards, and transaction protocols. As algorithmic fairness and interpretability emerge as first-class requirements, there is a need to develop generalized solutions that embed them as constraints and that work across a range of applications. In what follows, we highlight promising examples of our own recent and ongoing work that is motivated by this need. These examples underscore that tangible technical progress is possible and that much work remains to be done to offer systems support for the responsible management of the ADS life cycle. These examples are not intended to be exhaustive, but merely illustrate technical approaches that apply to different points of the data life cycle. Additional examples, and research directions, are discussed in Stoyanovich et al.[37] Before diving into the details, we recall the previously discussed mirror-reflection metaphor, as a reminder of the limits of technical interventions.

Data acquisition. Consider the use of an ADS for pre-screening employment applications. Historical underrepresentation of women and minorities in the workforce can lead to an underrepresentation of these groups in the training set, which in turn could push the ADS to reject more minority applicants or, more generally, to exhibit disparate predictive accuracy.[7] It is worth noting that the problem here is not only that some minorities are proportionally under-represented, but also that the absolute representation of some groups is low. Having 2% African Americans in the training set is a problem when they constitute 13% of the population. But it is also a problem to have only 0.2% Native Americans in the training set, even if that is representative of their proportion in the population. Such a low number can lead to Native Americans being ignored by the ADS as a small "outlier" group.

To mitigate low absolute representation, Asudeh et al.[2] assess the coverage of a given dataset over multiple categorical features. An important question for an ADS vendor is, then, what can it do about the lack of coverage. The proposed answer is to direct them to acquire more data, in a way that is cognizant of the cost of data acquisition. Asudeh et al.[2] use a threshold to determine an appropriate level of coverage and experimentally demonstrate an improvement in classifier accuracy for minority groups when additional data is acquired.

This work addresses a step in the ADS life cycle upstream from model training and shows how improving data representativeness can improve accuracy and fairness, in the sense of disparate predictive accuracy.[7] There are clear future opportunities to integrate coverage-enhancing interventions more closely into ADS life cycle management, both to help orchestrate the pipelines and, perhaps more importantly, to make data acquisition task-aware, setting coverage objectives based on performance requirements for the specific predictive analytics downstream rather than based on a global threshold.

Data preprocessing. Even when the acquired data satisfies representativeness requirements, it may still be subject to preexisting bias, as discussed in the "Preexisting bias" section. We may thus be interested in developing interventions to mitigate these effects. The algorithmic fairness community has developed dozens of methods for data and model de-biasing, yet the vast majority of these methods take an *associational interpretation of fairness* that is solely based on data, without reference to additional structure or context. In what follows, we present two recent examples of work that take a causal interpretation of fairness: a database repair framework for fair classification by Salimi et al.[29] and a framework for fair ranking that mitigates intersectional discrimination by Yang et al.[38] We focus on examples of causal fairness notions here because they correspond very closely to the methodological toolkit of data management by making explicit the use of structural information and constraints.

Causal fairness approaches—for example, Kilbertus et al.[21] and Kusner et al.[22]—capture background knowledge as causal relationships between variables, usually represented as causal DAGs, or directed acyclic graphs, in which nodes represent variables, and edges represent potential causal relationships. Consider the task of selecting job applicants at a moving company and the corresponding causal model in Figure 3, an example inspired by Datta et al.[10] Applicants are hired based on their qualification score Y, computed from weight-lifting ability X, and affected by gender G and race R, either directly or through X. By representing relationships between features in a causal DAG, we gain an ability to postulate which relationships between features and outcomes are legitimate and which are potentially discriminatory. In our example, the impact of gender (G) on the decision to hire an individual for a position with a moving company (Y) may be considered admissible if it flows through the node representing weight-lifting ability (X). On the other hand, the direct impact of gender on the decision to hire would constitute direct discrimination and would thus be considered inadmissible.

Salimi et al.[29] introduced a measure called *interventional fairness* for classification and showed how to achieve it based on observational data, without requiring the complete causal model. The authors consider the Markov boundary (MB)—parents, children, children's other parents—of a variable Y, which describes whether those nodes can potentially influence Y. Their key result is that the algorithm satisfies interventional fairness if the MB of the outcome is a subset of the MB of the admissible variables—that is, admissible variables "shield" the outcome from the influence of sensitive and inadmissible variables. This condition on the MB is used to design *database repair algorithms*, through a connection between the independence constraints encoding fairness and multivalued dependencies (MVD) that can be checked using the training data. Several repair algorithms are described, and the results show that in addition to satisfying interventional fairness, the classifier trained on repaired data performs well against associational fairness metrics.

The data management problems we are looking to address are not purely technical. Rather, they are socio-legal-technical.

As another example of a data preprocessing method that makes explicit use of structural assumptions, Yang et al.[38] developed a causal framework for *intersectionally fair ranking*. Their motivation is that it is possible to give the appearance of being fair with respect to each sensitive attribute, such as race and gender separately, while being unfair with respect to intersectional subgroups.[9] For example, if fairness is taken to mean proportional representation among the top-k, it is possible to achieve proportionality for each gender subgroup (for instance, men and women) and for each racial subgroup (for example, Black and White), while still having inadequate representation for a subgroup defined by the intersection of both attributes (for example, Black women). The gist of the methods of Yang et al.[38] is to use a causal model to compute model-based *counterfactuals*, answering the question: "What would this person's score be if she had been a Black woman (for example)?" and then ranking on counterfactual scores to achieve intersectional fairness.

Data-distribution debugging. We now return to our discussion of technical bias and consider data-distribution shifts, which may arise during data preprocessing and impact machine learning-model performance downstream. In contrast to important prior work on data-distribution shift detection in deployed models—for instance, Rabanser et al.[27]—our focus is explicitly on data manipulation, a cause of data-distribution shifts that has so far been overlooked. We will illustrate how this type of bias can arise and will suggest an intervention: a data-distribution debugger that helps surface technical bias, allowing a data scientist to mitigate it.[33]

Consider Ann, a data scientist at a job-search platform that matches profiles of job seekers with openings for which they are well-qualified and in which they may be interested. A job seeker's interest in a position is estimated based on several factors, including the salary and benefits being offered. Ann uses applicants' resumes, self-reported demographics, and employment histories as input. Following her company's best practices, she starts by splitting her dataset into training, validation, and test sets. Ann then uses pandas, scikit-learn, and accompanying data transformers to explore the data and implement data preprocessing, model selection, tuning, and validation. Ann starts preprocessing by computing value distributions and correlations for the features in the dataset and identifying missing values. She will use a default imputation method in scikit-learn to fill these in, replacing missing values with the mode value for that feature. Finally, Ann implements model selection and hyperparameter tuning, selecting a classifier that displays sufficient accuracy.

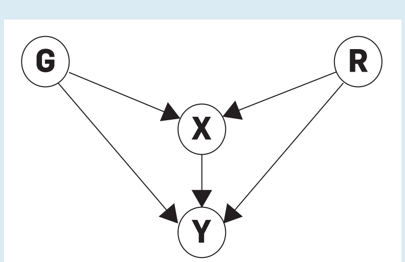
When Ann more closely considers the performance of the classifier, she observes a disparity in predictive accuracy:[7] Accuracy is lower for older job seekers, who are frequently matched with lower-paying positions than they would expect. Ann now needs to understand why this is the case, whether any of her technical choices during pipeline construction contributed to this disparity, and what she can do to mitigate this effect.

It turns out that this issue was the result of a *data-distribution bug*—a shift in the values of a feature that is important for the prediction and that is the result of a technical choice during pre-processing. Here, that feature is the number of years of job experience. The bug was introduced because of Ann's assumption that the values of this feature are *missing at random* and because of her choice to use mode imputation, which is consistent with this assumption. In fact, values were missing more frequently for older job seekers: They would not enter a high value in "years of experience" because they might be afraid of age discrimination. This observation is consistent with the intuition that individuals are more likely to withhold information that may disadvantage them. Taken together, these two factors resulted in imputed years-of-experience values skewing lower, leading to a lower salary-requirement estimate and impacting older applicants more than younger ones.

Data-distribution bugs are difficult to catch. In part, this is because different pipeline steps are implemented using different libraries and abstractions, and the data representation often changes from relational data to matrices during data preparation. Further, preprocessing often combines relational operations on tabular data with estimator/transformer pipelines, a composable and nestable abstraction for combining operations on array data which originates from scikit-learn and is executed in a hard-to-debug manner with nested function calls.

Grafberger et al. designed and implemented mlinspect,[15] a lightweight data-distribution debugger that supports automated inspection of data-intensive pipelines to detect the accidental introduction of statistical bias and linting for best practices. The mlinspect library extracts logical query plans—modeled as DAGs of preprocessing operators—from pipelines that use popular libraries, such as pandas and scikit-learn, and combines relational operations and estimator/transformer pipelines. The library automatically instruments the code and traces the impact of operators on properties, such as the distribution of sensitive groups in the data. mlinspect is a necessary first step in what we hope will be a long line of work in collectively developing data-science best practices and the tooling to support their broad adoption. Much important work remains to allow us to start treating data as a first-class citizen in software development.

Figure 3. Causal model includes sensitive attributes: *G* (gender), *R* (race), *X* (weight-lifting ability), and *Y* (utility score).



Overseeing ADS

We are in the midst of a global trend to regulate the use of ADSs. In the EU, the General Data Protection Regulation (GDPR) offers individuals protections regarding the collection, processing, and movement of their personal data, and applies broadly to the use of such data by governments and private-sector entities. Regulatory activity in several countries outside of the EU, notably Japan and Brazil, is in close alignment with the GDPR. In the U.S., many major cities, a handful of states, and the Federal government are establishing task forces and issuing guidelines about responsible development and technology use. With its focus on data rights and data-driven decision-making, the GDPR is, without a doubt, the most significant piece of technology regulation to date, serving as a "common denominator" for the oversight of data collection and usage, both in the EU and worldwide. For this reason, we will discuss the GDPR in some depth in the remainder of this section.

The GDPR aims to protect the rights and freedoms of natural persons with regard to how their personal data is processed, moved, and exchanged (Article 1). The GDPR is broad in scope and applies to "the processing of personal data wholly or partly by automated means" (Article 2), both in the private and public sectors. Personal data is broadly construed and refers to any information relating to an identified or identifiable natural person, called the *data subject* (Article 4). The GDPR aims to give data subjects insight into, and control over, the collection and processing of their personal data. Providing such insight, in response to the "right to be informed," requires technical methods for interpretability, discussed in the following section, "Interpretability for a range of stakeholders." We will also highlight, in the upcoming section, "Removing personal data," the right to erasure as a representative example of a regulatory requirement that raises a concrete data-management challenge. Additional details can be found in Abiteboul and Stoyanovich.[1]

As we have done throughout this article, we highlight specific challenges within the broad topic of ADS oversight and outline promising directions for technical work to address these challenges. It is important to keep in mind that ADS oversight will not admit a purely technical solution. Rather, we hope that technical interventions will be part of a robust distributed infrastructure of accountability, in which multiple stakeholder groups participate in ADS design, development, and oversight.

Interpretability for a range of stakeholders. Interpretability—allowing people to understand the process and decisions of an ADS—is critical to the responsible use of these systems. Interpretability means different things to different stakeholders, yet the common theme is that it allows people, including software developers, decision-makers, auditors, regulators, individuals who are affected by ADS decisions, and members of the public at large, to exercise agency by accepting or challenging algorithmic decisions and, in the case of decision-makers, to take responsibility for these decisions.

Interpretability rests on making explicit the interactions between the computational process and the data on which it acts. Understanding how code and data interact is important both when an ADS is interrogated for bias and discrimination, and when it is asked to explain an algorithmic decision that affects an individual.

To address the interpretability needs of different stakeholders, several recent projects have been developing tools based on the concept of a nutritional label—drawing an analogy to the food industry, where simple, standard labels convey information about ingredients and production processes. Short of setting up a chemistry lab, a food consumer would otherwise have no access to this information.

Figure 4. Ranking Facts for the CS department's dataset.

Similarly, consumers of data products or individuals affected by ADS decisions cannot be expected to reproduce the data collection and computational procedures. These projects include the Dataset Nutrition Label,[18] Datasheets for Datasets,[13] Model Cards,[25] and Ranking Facts,[39] which all use specific kinds of metadata to support interpretability. Figure 4 offers an example of a nutritional label; it presents Ranking Facts[39] to explain a ranking of computer science departments.

In much of this work, nutritional labels are manually constructed, and they describe a single component in the data life cycle, typically a dataset or a model. Yet, to be broadly applicable, and to faithfully represent the computational process and the data on which it acts, nutritional labels should be generated *automatically* or *semiautomatically* as a side effect of the computational process itself, embodying the paradigm of *interpretability by design*.[36] This presents an exciting responsible data-management challenge.

The data-management community has been studying systems and standards for metadata and provenance for decades.[17] This includes work on fine-grained provenance, where the goal is to capture metadata associated with a data product and propagate it through a series of transformations, to explain its origin and history of derivation, and to help answer questions about the robustness of the computational process and the trustworthiness of its results. There is now an opportunity to revisit many of these insights and to extend them to support the interpretability needs of different stakeholders, both technical and non-technical.

Removing personal data. The right to be forgotten is originally motivated by the desire of individuals not to be perpetually stigmatized by something they did in the past. Under pressure from despicable social phenomena such as revenge porn, it was turned into law in 2006 in Argentina, and since then in the EU, as part of the GDPR (Article 17), stating that data subjects have the right to request the timely erasure of their personal data.

An important technical issue of clear relevance to the data-management community is deletion of information in systems that are designed explicitly to accumulate data. Making data-processing systems GDPR-compliant has been identified as one of the data-management community's key research challenges.[35] The requirement of efficient deletion is in stark contrast with the typical requirements for data-management systems, necessitating substantial rethinking and redesign of the primitives, such as enhancing fundamental data structures with efficient delete operations.[30]

Data deletion must be both permanent and deep, in the sense that its effects must propagate through data dependencies. To start, it is difficult to guarantee that all copies of every piece of deleted data have actually been deleted. Further, when some data is deleted, the remaining database may become inconsistent, and may, for example, include dangling pointers. Additionally, production systems typically do not include a strong provenance mechanism, so they have no means of tracking the use of an arbitrary data item (one to be deleted) and reasoning about the dependencies on that data item in derived data products. Although much of the data-management community's attention over the years has been devoted to tracking and reasoning about provenance, primarily in relational contexts and in workflows (see Herschel et al.[17] for a recent survey), there is still important work to be done to make these methods both practically feasible and sufficiently general to accommodate current legal requirements.

An important direction that has only recently come into the academic community's focus concerns ascertaining the effects of a deletion on downstream processes that are not purely relational but include other kinds of data analysis tasks, such as data mining or predictive analytics. Recent research[14,31] argues that it is not sufficient to merely delete personal user data from primary data stores such as databases, but that machine-learning models trained on stored data also fall under the regulation. This view is supported by Recital 75 of the GDPR: "The risk to the rights and freedoms of natural persons ... may result from personal data processing ... where personal aspects are evaluated, in particular analyzing or predicting aspects concerning performance at work, economic situation, health, personal preferences or interests, reliability or behavior, location or movements." The machine-learning community has been working on this issue under the umbrella of *machine unlearning*.[6,14] Given a model, its training data, and a set of user data to delete/unlearn, the community proposes efficient ways to accelerate the retraining of the model. However, these approaches ignore the constraints imposed by the complexity of production set-ups (such as redeployment costs) and are thereby hard to integrate into real-world ML applications.[32]

We must learn to step outside our engineering comfort zone and to start reasoning in terms of values and beliefs.

Requests for deletion may also conflict with other laws, such as requirements to keep certain transaction data for some period or requirements for fault tolerance and recoverability. Understanding the impact of deletion requests on our ability to offer guarantees on system resilience and performance, and developing appropriate primitives and protocols for practical use, is another call to action for the data-management community.

Conclusion

In this article, we offered a perspective on the role that the data-management research community can play in the responsible design, development, use, and oversight of ADSs. We grounded our discussion in automated hiring tools, a specific use case that gave us ample opportunity to appreciate the potential benefits of data science and AI in an important domain and to get a sense of the ethical and legal risks.

An important point is that we cannot fully automate responsibility. While some of the duties of carrying out the task of, say, legal compliance can in principle be assigned to an algorithm, accountability for the decisions being made by an ADS always rests with a person. This person may be a decision-maker or a regulator, a business leader or a software developer. For this reason, we see our role as researchers in helping build systems that "expose the knobs" or responsibility to people.

Those of us in academia have an

additional responsibility to teach students about the social implications of the technology they build. Typical students are driven to develop technical skills and have an engineer's desire to build useful artifacts, such as a classification algorithm with low error rates. They are also increasingly aware of historical discrimination that can be reinforced, amplified, and legitimized with the help of technical systems. Our students will soon become practicing data scientists, influencing how technology companies impact society. It is our responsibility as educators to equip them with the skills to ask and answer the hard questions about the choice of a dataset, a model, or a metric. It is critical that the students we send out into the world understand responsible data science.

Toward this end, we are developing educational materials and teaching courses on responsible data science. H.V. Jagadish launched the first Data Science Ethics MOOC on the EdX platform in 2015. This course has since been ported to Coursera and FutureLearn, and it has been taken by thousands of students worldwide. Individual videos are licensed under Creative Commons and can be freely incorporated in other courses where appropriate. Julia Stoyanovich teaches highly visible technical courses on Responsible Data Science,[24] with all materials publicly available online. These courses are accompanied by a comic book series, developed under the leadership of Falaah Arif Khan, as supplementary reading.

In a pre-course survey, in response to the prompt, "Briefly state your view of the role of data science and AI in society", one student wrote: "It is something we cannot avoid and therefore shouldn't be afraid of. I'm glad that as a data science researcher, I have more opportunities as well as more responsibility to define and develop this 'monster' under a brighter goal." Another student responded, "Data Science [DS] is a powerful tool and has the capacity to be used in many different contexts. As a responsible citizen, it is important to be aware of the consequences of DS/AI decisions and to appropriately navigate situations that have the risk of harming ourselves or others."

Acknowledgments
This work was supported in part by NSF Grants No. 1934464, 1934565, 1934405, 1926250, 1741022, 1740996, 1916505, by Microsoft, and by Ahold Delhaize. All content represents the opinion of the authors and is not necessarily shared or endorsed by their respective employers or sponsors.

References
1. Abiteboul, S. and Stoyanovich, J. Transparency, fairness, data protection, neutrality: Data management challenges in the face of new regulation. *J. of Data and Information Quality 11*, 3 (2019), 15:1–15:9.
2. Asudeh, A., Jin, Z., and Jagadish, H.V. Assessing and remedying coverage for a given dataset. In *35th IEEE International Conference on Data Engineering* (April 2019), 554–565.
3. Baeza-Yates, R. Bias on the web. *Communications of the ACM 61*, 6 (2018), 54–61.
4. Biessmann, F., Salinas, D., Schelter, S., Schmidt, P., and Lange, D. Deep learning for missing value imputation in tables with non-numerical data. In *Proceedings of the 27th ACM Intern. Conf. on Information and Knowledge Management* (2018), 2017–2025.
5. Bogen, M. and Rieke, A. Help wanted: An examination of hiring algorithms, equity, and bias. *Upturn* (2018).
6. Cauwenberghs, G. and Poggio, T. Incremental and decremental support vector machine learning. *NeurIPS* (2001), 409–415.
7. Chen, I., Johansson, F., and Sontag, D. Why is my classifier discriminatory? S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018*, 3543–3554.
8. Chouldechova, A. and Roth, A. A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM 63*, 5 (2020), 82–89.
9. Crenshaw, K. Demarginalizing the intersection of race and sex: A Black feminist critique of antidiscrimination doctrine, feminist theory and antiracist politics. *University of Chicago Legal Forum 1* (1989), 139–167.
10. Datta, A., Sen, S., and Zick, Y. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *IEEE Symposium on Security and Privacy* (May 2016), 598–617.
11. Friedler, S., Scheidegger, C., and Venkatasubramanian, S. The (im)possibility of fairness: Different value systems require different mechanisms for fair decision making. *Communications of the ACM 64*, 4 (2021), 136–143.
12. Friedman, B. and Nissenbaum, H. Bias in computer systems. *ACM Transactions on Information Systems 14*, 3 (1996), 330–347.
13. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J., Wallach, H., Daumé III, H., and Crawford, K. Datasheets for datasets. *CoRR* (2018), abs/1803.09010.
14. Ginart, A., Guan, M., Valiant, G., and Zou, J. Making AI forget you: Data deletion in machine learning. In *NeurIPS* (2019), 3513–3526.
15. Grafberger, S., Stoyanovich, J., and Schelter, S. Lightweight inspection of data preprocessing in native machine learning pipelines. In *11th Conf. on Innovative Data Sys. Research, Online Proceedings* (January 2021), http://www.cidrdb.org.
16. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. A survey of methods for explaining black box models. *ACM Computing Surveys 51*, 5 (2019), 93:1–93:42.
17. Herschel, M., Diestelkämper, R., and Ben Lahmar, H. A survey on provenance: What for? What form? What from? *VLDB Journal 26*, 6 (2017), 881–906.
18. Holland, S., Hosny, A., Newman, S., Joseph, J., and Chmielinski, K. The dataset nutrition label: A framework to drive higher data quality standards. *CoRR* (2018), abs/1805.03677.
19. Jagadish, H.V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J., Ramakrishnan, R., and Shahabi, C. Big data and its technical challenges. *Communications of the ACM 57*, 7 (2014), 86–94.
20. Kappelhof, J. Survey research and the quality of survey data among ethnic minorities. In *Total Survey Error in Practice*, Wiley (2017).
21. Kilbertus, N., Carulla, M., Parascandolo, G., Hardt, M., Janzing, D., and Schölkopf, B. Avoiding discrimination through causal reasoning. In *Advances in Neural Information Processing Systems* (2017), 656–666.
22. Kusner, M., Loftus, J., Russell, C., and Silva, R. Counterfactual fairness. I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, In *Advances in Neural Information Processing Systems 30:* (2017), 4066–4076.
23. Lehr, D. and Ohm, P. Playing with the data: What legal scholars should learn about machine learning. *UC Davis Law Review 51*, 2 (2017), 653–717.
24. Lewis, A. and Stoyanovich, J. Teaching responsible data science. *Intern. J. of Artificial Intelligence in Education* (2021).
25. Mitchell, M., et al. Model cards for model reporting. In *Proceedings of the Conf. on Fairness, Accountability, and Transparency 2019*, 220–229.
26. Olteanu, A., Castillo, C., Diaz, F., and Kiciman, E. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers Big Data 2*, 13 (2019).
27. Rabanser, S., Günnemann, S., and Lipton, Z. Failing loudly: An empirical study of methods for detecting dataset shift. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors. In *Advances in Neural Information Processing Systems 32* (December 2019), 1394–1406.
28. Reeves, R. and Halikias, D. Race gaps in SAT scores highlight inequality and hinder upward mobility. Brookings (2017), https://www.brookings.edu/research/race-gaps-in-sat-scores-highlight-inequality-and-hinder-upward-mobility.
29. Salimi, B., Rodriguez, L., Howe, B., and Suciu, D. Interventional fairness: Causal database repair for algorithmic fairness. P.A. Boncz, S. Manegold, A. Ailamaki, A. Deshpande, and T. Kraska, editors. In *Proceedings of the 2019 Intern. Conf. on Management of Data*, 793–810.
30. Sarkar, S., Papon, T., Staratzis, D., and Athanassoulis, M. Lethe: A tunable delete-aware LSM engine. In *Proceedings of the 2020 Intern. Conf. on Management of Data*.
31. Schelter, S. "Amnesia"–a selection of machine learning models that can forget user data very fast. *Conf. on Innovative Data Systems Research*, 2020.
32. Schelter, S., Grafberger, S., and Dunning, T. HedgeCut: Maintaining randomised trees for low-latency machine unlearning. In *Proceedings of the 2021 Intern. Conf. on Management of Data*.
33. Schelter, S. and Stoyanovich, J. Taming technical bias in machine learning pipelines. *IEEE Data Engineering Bulletin 43*, 4 (2020).
34. Selbst, A. Disparate impact in big data policing. *Georgia Law Review 52*, 109 (2017).
35. Shastri, S., Banakar, V., Wasserman, M., Kumar, A., and Chidambaram, V. Understanding and benchmarking the impact of GDPR on database systems. *PVLDB* (2020).
36. Stoyanovich, J. and Howe, B. Nutritional labels for data and models. *IEEE Data Engineering Bulletin 42*, 3 (2019), 13–23.
37. Stoyanovich, J., Howe, B., and Jagadish, H.V. Responsible data management. In *Proceedings of the VLDB Endowment 13*, 12 (2020), 3474–3488.
38. Yang, K., Loftus, J., and Stoyanovich, J. Causal intersectionality and fair ranking. K. Ligett and S. Gupta, editors. In *2nd Symposium on Foundations of Responsible Computing, Volume 192 of LIPICS, Schloss Dagstuhl–Leibniz Center for Informatics* (June 2021), 7:1-7:20.
39. Yang, K., Stoyanovich, J., Asudeh, A., Howe, B., Jagadish, H.V., and Miklau, G. A nutritional label for rankings. G. Das, C. Jermaine, and P. Bernstein, editors. In *Proceedings of the 2018 Intern. Conf. on Management of Data*, 1773–1776.
40. Zehlike, M., Yang, K., and Stoyanovich, J. Fairness in ranking: A survey. *CoRR* (2021), abs/2103.14000.

Julia Stoyanovich (stoyanovich@nyu.edu) is an associate professor at New York University, New York, NY, USA.

Serge Abiteboul is a researcher at Inria & École Normale Supérieure, Paris, France.

Bill Howe is an associate professor at the University of Washington, Seattle, WA, USA.

H.V. Jagadish is a professor at the University of Michigan, Ann Arbor, MI, USA.

Sebastian Schelter is an assistant professor at the University of Amsterdam, Amsterdam, The Netherlands.

DOI:10.1145/3528086

Improving the peer review process in a scientific manner shows promise.

BY NIHAR B. SHAH

Challenges, Experiments, and Computational Solutions in Peer Review

PEER REVIEW IS a cornerstone of scientific research. Although quite ubiquitous today, peer review in its current form became popular only in the middle of the 20th century. Peer review looks to assess research in terms of its competence, significance, and originality.[6] It aims to ensure quality control to reduce misinformation and confusion[4] thereby upholding the integrity of science and the public trust in science.[49] It also helps in improving the quality of the published research.[17] In the presence of an overwhelming number of papers written, peer review also has another role:[40] "Readers seem to fear the firehose of the Internet: they want somebody to select, filter, and purify research material."

Surveys[48] of researchers in several scientific fields find that peer review is highly regarded by most researchers. Indeed, most researchers believe peer review gives confidence in the academic rigor of published articles and that it improves the quality of the published papers. These surveys also find there is a considerable and increasing desire for improving the peer-review process.

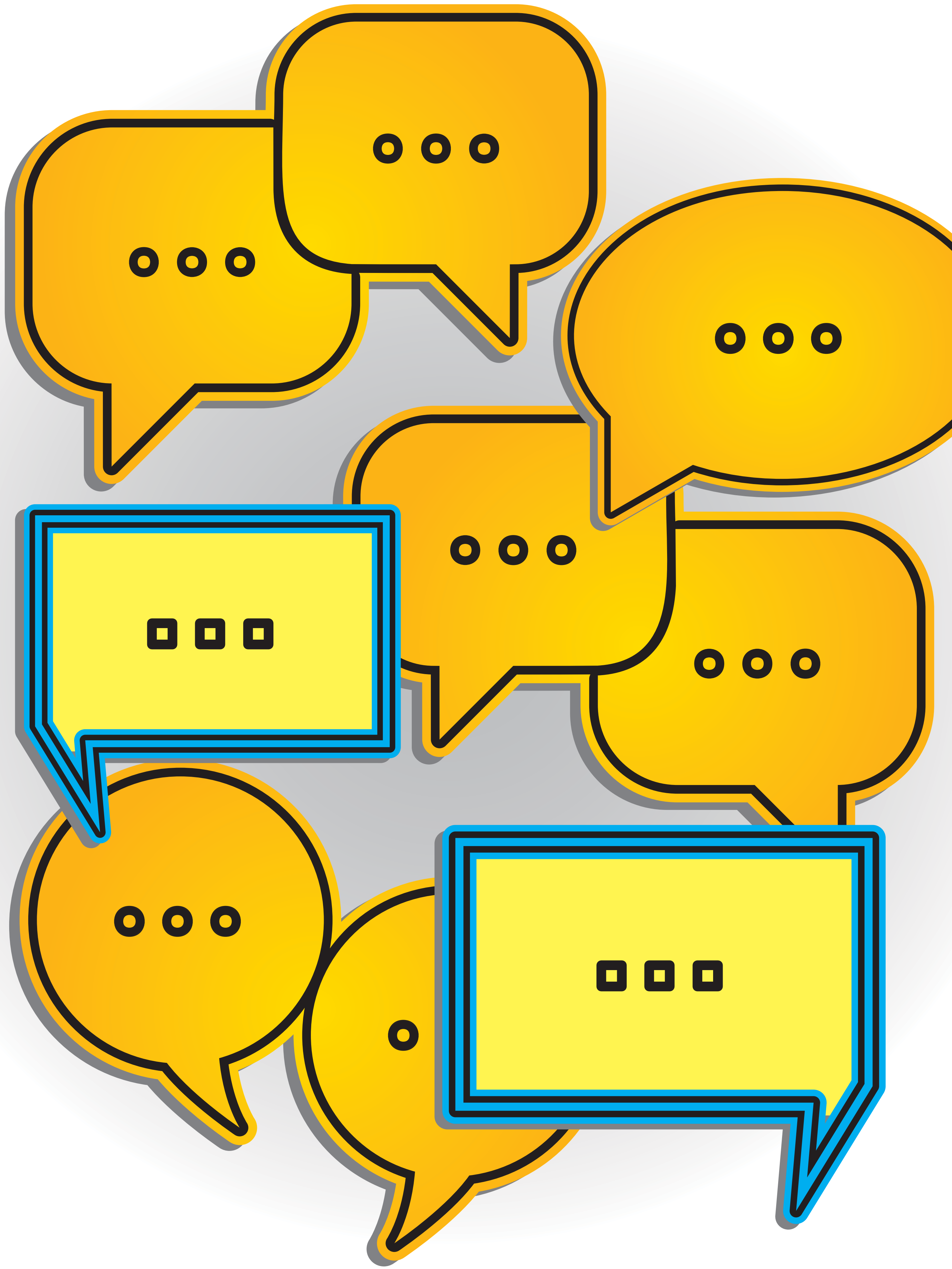
Peer review is assumed to provide a "mechanism for rational, fair, and objective decision making."[17] For this, one must ensure evaluations are "independent of the author's and reviewer's social identities and independent of the reviewer's theoretical biases and tolerance for risk."[22] There are, however, key challenges toward these goals. The following quote from Rennie[35] summarizes many of the challenges in peer review: *"Peer review is touted as a demonstration of the self-critical nature of science. But it is a human system. Everybody involved brings prejudices, misunderstandings, and gaps in knowledge, so no one should be surprised that peer review is often biased and inefficient. It is occasionally corrupt, sometimes a charade, an open temptation to plagiarists. Even with the best of intentions, how and whether peer review identifies high-quality science is unknown. It is, in short, unscientific."* Problems in peer review have consequences much beyond the outcome for a specific paper or grant proposal, particularly due to the widespread prevalence of the Matthew effect ("rich get richer") in academia.

In this article, we discuss several manifestations of the aforementioned challenges, experiments that help understand these issues and the trade-offs involved, and various (computational) solutions in the literature. For concrete-

» key insights

- **In computer science, the design of computational tools to assist peer review and experiments to understand various trade-offs form a fast-growing area of research. This line of research has already made significant impact with various computational tools widely deployed and experiments informing review policies.**
- **Further progress on these open problems can have substantial impact on peer review as well as on many other applications involving distributed human evaluations.**

IMAGERY BY CREATIVE ICON STYLES; COMPOSITION BY ANDRIJ BORYS ASSOCIATES



ness, our exposition focuses on peer review in scientific conferences. Most points discussed also apply to other forms of peer review such as review of grant proposals (used to award billions of dollars' worth of grants every year), journal review, and peer evaluation of employees in organizations. Moreover, any progress on this topic has implications for a variety of applications such as crowdsourcing, peer grading, recommender systems, hiring, college admissions, judicial decisions, and healthcare. The common thread across these applications is they involve distributed human evaluations: a set of people need to evaluate a set of items, but every item is evaluated by a small subset of people and every person evaluates only a small subset of items.

An Overview of the Review Process

We begin with an overview of a representative conference review process. The process is coordinated on an online platform known as a conference management system. Each participant in the peer review process has one or more of the following four roles: program chairs, who coordinate the entire peer-review process; authors, who submit papers to the conference; reviewers, who read the papers and provide feedback and evaluations; and meta reviewers, who are intermediaries between reviewers and program chairs.

Authors must submit their papers by a predetermined deadline. The submission deadline is immediately followed by "bidding," where reviewers can indicate which papers they are willing or unwilling to review. The papers are then assigned to reviewers for review. Each paper is reviewed by a handful of (typically three to six) reviewers. The number of papers per reviewer varies across conferences and can range from a handful (three to eight in the field of artificial intelligence) to a few dozen papers. Each meta reviewer is asked to handle a few dozen papers, and each paper is handled by one meta reviewer.

Each reviewer is required to provide reviews for their assigned papers before a set deadline. The reviews comprise an evaluation of the paper and suggestions to improve the paper. The authors may then provide a rebuttal to the review, which could clarify any inaccuracies or misunderstandings in the reviews. Reviewers are asked to read the authors' rebuttal (as well as other reviews) and update their reviews accordingly. A discussion for each paper then takes place between its reviewers and meta reviewer. Based on all this information, the meta reviewer then recommends to the program chairs a decision about whether to accept the paper to the conference. The program chairs eventually make the decisions on all papers.

While this description is representative of many conferences (particularly large conferences in the field of artificial intelligence), individual conferences may have some deviations. For example, many smaller-sized conferences do not have meta reviewers, and the final decisions are made via an in-person or online discussion between the entire pool of reviewers and program chairs. That said, most of the content to follow in this article is applicable broadly.

Mismatched Reviewer Expertise

The assignment of the reviewers to papers determines whether reviewers have the necessary expertise to review a paper. Time and again, a top reason for authors to be dissatisfied with reviews is the mismatch of the reviewers' expertise with the paper. For small conferences, the program chairs may assign reviewers themselves. However, this approach does not scale to conferences with hundreds or thousands of papers. As a result, reviewer assignments in most moderate-to-large-sized conferences are performed in an automated manner (sometimes with a bit of manual tweaking). There are two stages in the automated assignment procedure.

Computing similarity scores. The first stage of the assignment process involves computing a "similarity score" for every reviewer-paper pair. The similarity score $S_{p,r}$ between any paper p and any reviewer r is a number between 0 and 1 that captures the expertise match between reviewer r and paper p. A higher similarity score means a better-envisaged quality of the review. The similarity is computed based on one or more of the following sources of data.

Subject-area selection. When sub-

The outcomes of peer review can have a considerable influence on the career trajectories of authors. While we believe most participants in peer review are honest, the stakes can unfortunately incentivize dishonest behavior.

mitting a paper, authors are required to indicate one or more subject areas to which the paper belongs. Before the review process begins, each reviewer also indicates one or more subject areas of their expertise. Then, for every paper-reviewer pair, a score is computed as the amount of intersection between the paper's and reviewer's chosen subject areas.

Text matching. The text of the reviewer's previous papers is matched with the text of the submitted papers using natural language processing techniques. We summarize a couple of approaches here.[9,29] One approach is to use a language model. At a high level, this approach assigns a higher text-score similarity if (parts of) the text of the submitted paper has a higher likelihood of appearing in the corpus of the reviewer's previous papers under an assumed language model. A simple incarnation of this approach assigns a higher text-score similarity if the words that (frequently) appear in the submitted paper also appear frequently in the papers in the reviewer's previous papers.

A second common approach uses "topic modeling." Each paper or set of papers is converted to a vector. Each coordinate of this vector represents a topic that is extracted in an automated manner from the entire set of papers. For any paper, the value of a specific coordinate indicates the extent to which the paper's text pertains to the corresponding topic. The text-score similarity is the dot product of the submitted paper's vector and a vector corresponding to the reviewer's past papers.

The design of algorithms to compute similarities more accurately through advances in natural language processing is an active area of research.[32]

Bidding. Many conferences employ a "bidding" procedure where reviewers are shown the list of submitted papers and asked to indicate which papers they are willing or unwilling to review. A sample bidding interface is shown in Figure 1.

Cabanac and Preuss[7] analyze the bids made by reviewers in several conferences. Here, along with each review, the reviewer is also asked to report their confidence in their evaluation. They find that assigning papers for which reviewers have made positive (willing) bids is associated with higher confidence reported by reviewers for their reviews. This observation suggests the importance of assigning papers to reviewers who bid positively for the paper.

Many conferences suffer from the lack of adequate bids on a large fraction of submissions. For instance, 146 out of the 264 submissions at the ACM/IEEE Joint Conference on Digital Libraries (JCDL) 2005 had zero positive bids.[36] The Neural Information Processing Systems (NeurIPS) 2016 conference in the field of machine learning aimed to assign six reviewers and one meta-reviewer to each of the 2,425 papers, but 278 papers received at most two positive bids and 816 papers received at most five positive bids from reviewers, and 1,019 papers received zero positive bids from meta reviewers.[38]

Cabanac and Preuss[7] also uncover a problem with the bidding process. The conference management systems there assigned each submitted paper a number called a "paperID." The bidding interface then ordered the papers according to the paperIDs, that is, each reviewer saw the paper with the smallest paperID at the top of the list displayed to them and increasing paperIDs thereafter. They found that the number of bids placed on submissions generally decreased with an increase in the paperID value. This phenomenon is explained by well-studied serial-position effects[31] that humans are more likely to interact with an item if shown at the top of a list rather than down the list. Hence, this choice of interface results in a systematic bias against papers with greater values of assigned paper IDs.

Cabanac and Preuss suggest exploiting serial-position effects to ensure a better distribution of bids across papers by ordering the papers shown to any reviewer in increasing order of bids already received. However, this approach can lead to a high reviewer dissatisfaction since papers of the reviewer's interest and expertise may end up significantly down the list, whereas papers unrelated to the reviewer may show up at the top. An alternative ordering strategy used commonly in conference management systems today is to first compute a similarity between all reviewer-paper pairs using other data sources, and then order the papers in decreasing order of similarities with the reviewer. Although this approach addresses reviewer satisfaction, it does not exploit serial-position effects like the idea of Cabanac and Preuss. Moreover, papers with only moderate similarity with all reviewers (for example, if the paper is interdisciplinary) will not be shown at the top of the list to anyone. These issues motivate an algorithm[10] that dynamically orders papers for every reviewer by trading off reviewer satisfaction (showing papers with higher similarity at the top) with balancing paper bids (showing papers with fewer bids at the top).

Combining data sources. The data sources discussed above are then merged into a single similarity score. One approach is to use a specific formula for merging, such as $S_{p,r} = 2^{\text{bid-score}_{p,r}}(\text{subject-score}_{p,r} + \text{text-score}_{p,r})/4$ used in the NeurIPS 2016 conference.[38] A second approach involves program chairs trying out various combinations, eyeballing the resulting assignments, and picking the combination that seems to work best. Finally and importantly, if any reviewer r has a conflict with an author of any paper p (that is, if the reviewer is an author of the paper or is a colleague or collaborator of any author of the paper), then the similarity $S_{p,r}$ is set as –1 to ensure this reviewer is never assigned this paper.

Computing the assignment. The second stage assigns reviewers to papers in a manner that maximizes some function of the similarity scores of the assigned reviewer-paper pairs. The most popular approach is to maximize

Figure 1. A sample interface for bidding.

Papers:	Not Willing To Review	Indifferent	Eager To Review
Toward More Accurate NLP Models	○	○	○
Interpreting AI Decision Making	○	○	○
Multiagent Cooperative Board Games	○	○	○

the total sum of the similarity scores of all assigned reviewer-paper pairs:[9]

$$\underset{\text{assignment}}{\text{maximize}} \sum_{\text{papers } p} \sum_{\substack{\text{reviewers } r \\ \text{assigned to paper } p}} S_{p,r},$$

subject to load constraints that each paper is assigned a certain number of reviewers and no reviewer is assigned more than a certain number of papers.

This approach of maximizing the sum of similarity scores can lead to unfairness to certain papers.[42] As a toy example illustrating this issue, consider a conference with three papers and six reviewers, where each paper is assigned one reviewer and each reviewer is assigned two papers. Suppose the similarities are given by the table on the left-hand side of Figure 2. Here {paper A, reviewer 1, reviewer 2} belong to one research discipline, {paper B, reviewer 3, reviewer 4} belong to a second research discipline, and paper C's content is split across these two disciplines. Maximizing the sum of similarity scores results in the assignment shaded light/orange in the left-hand side of Figure 2. Observe that the assignment for paper C is quite poor: all assigned reviewers have a zero similarity. This is because this method assigns better reviewers to papers A and B at the expense of paper C. Such a phenomenon is indeed found to occur in practice. The paper[18] analyzes data from the Computer Vision and Pattern Recognition (CVPR) 2017 and 2018 conferences, which have several thousand papers. The analysis reveals there is at least one paper each to which this method assigns all reviewers with a similarity score of zero, whereas other assignments can ensure that every paper has at least some reasonable reviewers.

The right-hand side of Figure 2 depicts the same similarity matrix. The cells shaded light/blue depict an alternative assignment. This assignment is more balanced: it assigns papers A and B reviewers of lower similarity as compared to earlier, but paper C now has reviewers with a total similarity of 1 rather than 0. This assignment is an example of an alternative approach[13,18,42] that optimizes for the paper which is worst-off in terms of the similarities of its assigned reviewers:

$$\underset{\text{assignment}}{\text{maximize}} \ \underset{\text{papers } p}{\text{minimum}} \sum_{\substack{\text{reviewers } r \\ \text{assigned to paper } p}} S_{p,r},$$

The approach then optimizes for the paper that is the next worst-off and so on. Evaluations[18,42] of this approach on several conferences reveal it significantly mitigates the problem of imbalanced assignments, with only a moderate reduction in the sum-similarity score value as compared to the approach of maximizing sum-similarity scores.

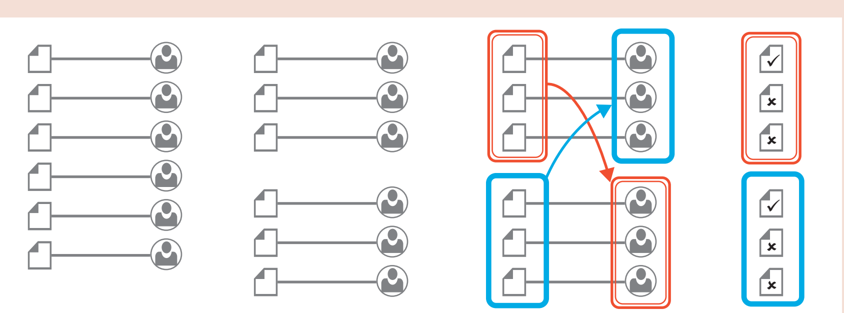
Recent work also incorporates various other desiderata in the reviewer-paper assignments.[23] An emerging concern when doing the assignment is that of dishonest behavior.

Figure 2. Assignment in a fictitious example conference using the popular sum-similarity optimization method (left) and a more balanced approach (right).

	Paper A	Paper B	Paper C
Reviewer 1	0.9	0	0.5
Reviewer 2	0.6	0	0.5
Reviewer 3	0	0.9	0.5
Reviewer 4	0	0.6	0.5
Reviewer 5	0	0	0
Reviewer 6	0	0	0

	Paper A	Paper B	Paper C
Reviewer 1	0.9	0	0.5
Reviewer 2	0.6	0	0.5
Reviewer 3	0	0.9	0.5
Reviewer 4	0	0.6	0.5
Reviewer 5	0	0	0
Reviewer 6	0	0	0

Figure 3. Partition-based method for strategyproofness.



Dishonest Behavior

The outcomes of peer review can have a considerable influence on the career trajectories of authors. While we believe that most participants in peer review are honest, the stakes can unfortunately incentivize dishonest behavior. We discuss two such issues.

Lone wolf. Conference peer review is competitive, that is, a roughly pre-determined number (or fraction) of submitted papers are accepted. Moreover, many authors are also reviewers. Thus, a reviewer could increase the chances of acceptance of their own papers by manipulating the reviews (for example, providing lower ratings) for other papers.

A controlled study by Balietti et al.[3] examined the behavior of participants in competitive peer review. Participants were randomly divided into two conditions: one where their own review did not influence the outcome of their own work, and the other where it did. Balietti et al. observed that the ratings given by the latter group were drastically lower than those given by the former group. They concluded that "*competition incentivizes reviewers to behave strategically, which reduces the fairness of evaluations and the consensus among referees.*" The study also found that the number of such strategic reviews increased over time, indicating a retribution cycle in peer review.

This motivates the requirement of "strategyproofness": no reviewer must be able to influence the outcome of their own submitted paper by manipulating the reviews they provide. A simple yet effective idea to ensure strategyproofness is called the partition-based method.[1] The key idea of the partition-based method is illustrated in Figure 3. Consider the "authorship" graph in Figure 3a whose vertices comprise the submitted papers and reviewers, and an edge exists between a paper and reviewer if the reviewer is an author of that paper. The partition-based method first partitions the reviewers and papers into two (or more) groups such that all authors of any paper are in the same group as the paper (Figure 3b). Each paper is then assigned for review to reviewers in the other group(s) (Figure 3c). Finally, the decisions for the papers in any group are made independent of the other group(s) (Figure 3d). This method is strategy-proof since any reviewer's reviews influence only papers in other groups, whereas the reviewer's own authored papers belong to the same group as the reviewer.

The partition-based method is largely studied in the context of peer-grading-like settings. In peer grading, one may assume each paper (homework) is authored by one reviewer (student) and each reviewer authors one paper, as is the case in Figure 3. Conference peer review is more complex: papers have multiple authors and authors submit multiple papers. Consequently, in conference peer review it is not clear if there even exists a partition. Even if such a partition exists, the partition-based constraint on the assignment could lead to a considerable reduction in the assignment quality. Such questions about realizing the partition-based method in conference peer review are still open, with promising initial results[51] showing that such partitions do exist in practice and the reduction in quality of assignment may not be too drastic.

Coalitions. Several recent investigations have uncovered dishonest coalitions in peer review.[24,46] Here a reviewer and an author come to an understanding: the reviewer manipulates the system to try to be assigned the author's paper, then accepts the paper if assigned, and the author offers quid pro quo either in the same conference or elsewhere. There may be coalitions between more than two people, where a group of reviewers (who are also authors) illegitimately push for each other's papers.

The first line of defense against such behavior is conflicts of interest: one may suspect that colluders may know each other well enough to also have co-authored papers. Then treating previous coauthor-ship as a conflict of interest and ensuring to not assign any paper to a reviewer who has a conflict with its authors, may seem to address this problem. It turns out that even if colluders collaborate, they may go to great lengths to enable dishonest behavior:[46] *"There is a chat group of a few dozen authors who in subsets work on common topics and carefully ensure not to co-author any papers with each other so as to keep out of each other's conflict lists (to the extent that even if there is collaboration they voluntarily give up authorship on one paper to prevent conflicts on many future papers)."*

Biases with respect to author identities are widely debated in computer science.

A second line of defense addresses attacks where two or more reviewers (who have also submitted their own papers) aim to review each other's papers. This has motivated the design of assignment algorithms[14] with an additional constraint of disallowing any loops in the assignment, that is, ensuring to not assign two people each other's' papers. This defense prevents colluders engaging in a quid pro quo in the same venue. However, this defense can be circumvented by colluders who avoid forming a loop, for example, where a reviewer helps an author in a certain conference and the author reciprocates elsewhere. Moreover, it has been uncovered that, in some cases, an author pressures a certain reviewer to get assigned and accept a paper.[19] This line of defense does not guard against such situations where there is no quid pro quo within the conference.

A third line of defense is based on the observation that the bidding stage of peer review is perhaps the most easily manipulable: reviewers can significantly increase the chances of being assigned a paper they may be targeting by bidding strategically.[16,50] This suggests curtailing or auditing bids, and this approach is followed in the paper.[50] This work uses the bids from all reviewers as labels to train a machine learning model that predicts bids based on the other sources of data. This model can then be used as the similarities for making the assignment. It thereby mitigates dishonest behavior by de-emphasizing bids that are significantly different from the remaining data.

Dishonest collusions may also be executed without bidding manipulations. For example, the reviewer/paper subject areas and reviewer profiles may be strategically selected to increase the chances of getting assigned the target papers.

Security researchers have demonstrated the vulnerability of paper assignment systems to attacks where an author could manipulate the PDF (portable document format) of their submitted paper so that a certain reviewer gets assigned.[27] These attacks insert text in the PDF of the submitted paper in a manner that satisfies three properties: the inserted text matches keywords from a target reviewers' paper; this text is not visible to the human reader; and this text is read by the (automated) parser which computes the text-similarity-score between the submitted paper and the reviewer's past papers. These properties guarantee a high similarity for the colluding reviewer-paper pair, while ensuring that no human reader detects it. These attacks are accomplished by targeting the font embedding in the PDF, as illustrated in Figure 4. Empirical evaluations on the reviewer-assignment system used at the International Conference on Computer Communications (INFOCOM) demonstrated the high efficacy of these attacks by being able to get papers matched to target reviewers. In practice, there may be other attacks used by malicious participants beyond what program chairs and security researchers have detected to date.

In some cases, the colluding reviewers may naturally be assigned to the target papers without any manipulation of the assignment process:[46] *"They exchange papers before submissions and then either bid or get assigned to review each other's papers by virtue of having expertise on the topic of the papers."*

The final defense[16] discussed here makes no assumptions on the nature of manipulation and uses randomized assignments to mitigate the ability of participants to conduct such dishonest behavior. Here, the program chairs specify a value between 0 and 1. The randomized assignment algorithm chooses the best possible assignment subject to the constraint that the probability of assigning any reviewer to any paper be at most that value. The upper bound on the probability of assignment leads to a higher chance that an independent reviewer will be assigned to any paper, irrespective of the manner or magnitude of manipulations by dishonest reviewers. Naturally, such a randomized assignment may also preclude honest reviewers with appropriate expertise from getting assigned. Consequently, the program chairs can choose the probability values at run time by inspecting the trade-off between the amount of randomization and the quality of the assignment (Figure 5). This defense was used in the Advancement of Artificial Intelligence (AAAI) 2022 conference.

The recent discoveries of dishonest behavior also pose important questions of law, policy, and ethics for dealing with such behavior: How should program chairs deal with suspicious behavior, and what constitutes appropriate penalties? A case that led to widespread debate is an ACM investigation that banned certain guilty parties from participating in ACM venues for several years without publicly revealing the names of all guilty parties. Furthermore, some conferences only impose the penalty of rejection of a paper if an author is found to indulge in dishonest behavior including blatant plagiarism. This raises concerns of lack of transparency, and that guilty parties may still participate and possibly continue dishonest behavior in other conferences or grant reviews.

Miscalibration

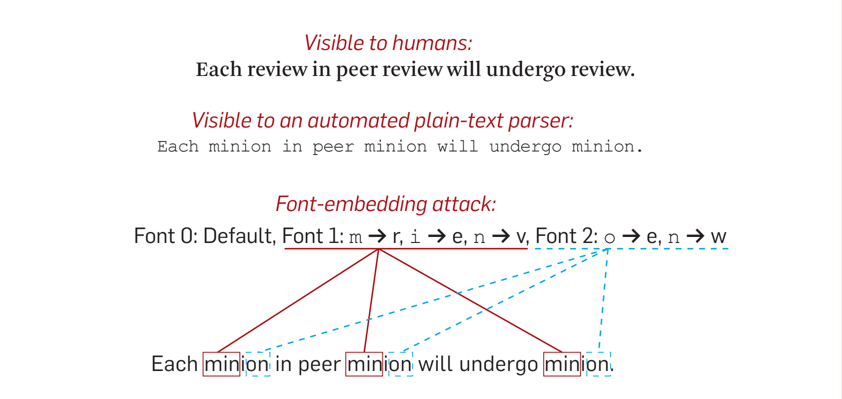
Reviewers are often asked to provide assessments of papers in terms of ratings, and these ratings form an integral part of the final decisions. However, it is well known[12,30,39] that the same rating may have different meanings for different individuals: *"A raw rating of 7 out of 10 in the absence of any other information is potentially useless."*[30] In the context of peer review, some reviewers are lenient and generally provide high ratings whereas some others are strict and rarely give high ratings; some reviewers are more moderate and tend to give borderline ratings whereas others provide ratings at the extremes, and so on.

Miscalibration causes arbitrariness and unfairness in the peer review process:[39] *"the existence of disparate categories of reviewers creates the potential for unfair treatment of authors. Those whose papers are sent by chance to assassins/demoters are at an unfair disadvantage, while zealots/pushovers give authors an unfair advantage."*

Miscalibration may also occur if there is a mismatch between the conference's overall expectations and

Figure 4. An attack on the assignment system via font embedding in the PDF of the submitted paper.[27]

Suppose the colluding reviewer has the word "minion" as most frequently occurring in their previous papers, whereas the paper submitted by the colluding author has "review" as most commonly occurring. The author creates two new fonts that map the plain text to rendered text as shown. The author then chooses fonts for each letter in the submitted paper in such a manner that the word "minion" in plain text renders as "review" in the PDF. A human reader will now see "review," but an automated parser will read "minion." The submitted paper will then be assigned to the target reviewer by the assignment system, whereas no human reader will see "minion" in the submitted paper.



reviewers' individual expectations. As a concrete example, the NeurIPS 2016 conference asked reviewers to rate papers on a scale of 1 through 5 (where 5 is best) and specified an expectation regarding each value on the scale. However, there was a significant difference between the expectations and the ratings given by reviewers.[38] For instance, the program chairs asked reviewers to give a rating of 3 or better if the reviewer considered the paper to lie in the top 30% of all submissions, but the actual number of reviews with the rating 3 or better was nearly 60%.

There are two popular approaches toward addressing the problem of miscalibration of individual reviewers. The first approach[11,37] is to make simplifying assumptions on the nature of the miscalibration, for instance, assuming that miscalibration is linear or affine. Most works taking this approach assume that each paper p has some "true" underlying rating θ_p, that each reviewer r has two "miscalibration parameters" $a_r > 0$ and b_r, and that the rating given by any reviewer r to any paper p is given by $a_r\theta_p + b_r$ + noise. These algorithms then use the ratings to estimate the "true" paper ratings θ, and possibly also reviewer parameters

The simplistic assumptions described here are frequently violated in the real world.[5] Algorithms based on such assumptions were tried in some conferences, but based on manual inspection by the program chairs, were found to perform poorly.

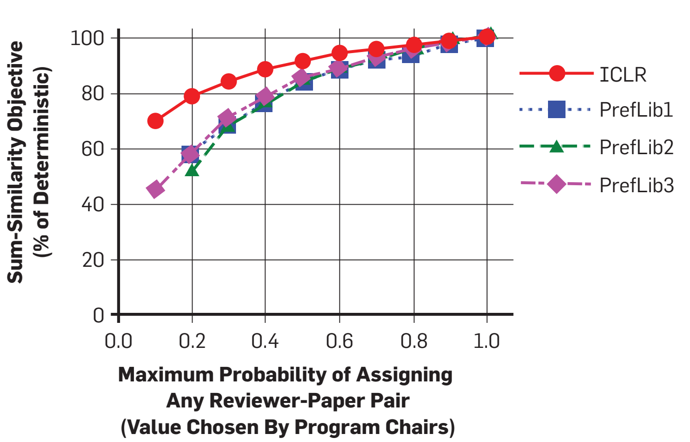
A second popular approach[12,30] toward handling miscalibrations is via rankings: either ask reviewers to give a ranking of the papers they are reviewing (instead of providing ratings), or alternatively, use the rankings obtained by converting any reviewer's ratings into a ranking of their reviewed papers. Using rankings instead of ratings *"becomes very important when we combine the rankings of many viewers who often use completely different ranges of scores to express identical preferences."*[12]

Ratings can provide some information even in isolation. It was shown recently[47] that even if the miscalibration is arbitrary or adversarially chosen, unquantized ratings can yield better results than rankings alone. Rankings also have their benefits. In NeurIPS 2016, out of all pairs of papers reviewed by the same reviewer, the reviewer gave an identical rating to both papers for 40% of the pairs.[38] In such situations, rankings can help break ties among these papers, and this approach was followed in the International Conference on Machine Learning (ICML) 2021. A second benefit of rankings is to check for possible inconsistencies. For instance, the NeurIPS 2016 conference elicited rankings from reviewers on an experimental basis. They then compared these rankings with the ratings given by the reviewers. They found that 96 (out of 2,425) reviewers had rated some paper as strictly better than another on all four criteria but reversed the pair in the overall ranking.[38]

Addressing miscalibration in peer review is a wide-open problem. The small per-reviewer sample sizes due to availability of only a handful of reviews per reviewer is a key obstacle: for example, if a reviewer reviews just three papers and gives low ratings, it is difficult to infer from this data as to whether the reviewer is generally strict. This impediment calls for designing protocols or privacy-preserving algorithms that allow conferences to share some reviewer-specific calibration data with one another to calibrate better.

Figure 5. Trading off the quality of the assignment (sum similarity on *y*-axis) with the amount of randomness (value specified by program chairs on *x*-axis) to mitigate dishonest coalitions.[16]

The similarity scores for the "ICLR" plot are reconstructed[51] via text-matching from the International Conference on Learning Representations (ICLR conference) 2018 which had 911 submissions. The "Preflib" plots are computed on bidding data from three small-sized conferences (with 54, 52, and 176 submissions), obtained from the Preflib database.[28]



Subjectivity

We discuss two challenges in peer review pertaining to reviewers' subjective preferences that hamper the objectivity of peer review.

Commensuration bias. Conference program chairs often provide criteria to reviewers for judging papers. However, different reviewers have different, subjective opinions about the relative importance of various criteria in judging papers. The overall evaluation of a paper then depends on the individual reviewer's preference on how to aggregate the evaluations on the individual criteria. This dependence on factors exogenous to the paper's content results in arbitrariness in the review process. On the other hand, to ensure fairness, all (comparable) papers should be judged by the same yardstick. This issue is known as "commensuration bias."[21]

For example, suppose three reviewers consider empirical performance of any proposed algorithm as most important, whereas most others highly regard novelty. Then a novel paper whose proposed algorithm has a modest empirical performance is rejected if reviewed by these three reviewers but would have been accepted by any other set of reviewers. The paper's fate thus depends on the subjective preference of the assigned reviewers.

The program chairs of AAAI 2013 conference recognized this problem of commensuration bias. With an admirable goal of ensuring a uniform policy of how individual criteria are aggregated into an overall recommendation across all papers and reviewers, they announced specific rules on how reviewers should aggregate their ratings on the eight criteria into an overall rating. The goal was commendable, but unfortunately, the proposed rules had shortcomings. For example,[33] on a scale of 1 to 6 (where 6 is best), one rule required giving an overall rating of "strong accept" if a paper received a rating of 5 or 6 for some criterion and did not get a 1 for any criteria. This may seem reasonable at first, but looking at it more carefully, it implies a strong acceptance for any paper that receives a 5 for the criterion of clarity but receives a low rating of 2 in every other criterion. More generally, specifying a set of rules for aggregation of 8 criteria amounts to specifying an 8-dimensional function, which can be challenging to craft by hand.

Due to concerns about commensuration bias, the NeurIPS 2016 conference did not ask reviewers to provide any overall ratings. NeurIPS 2016 instead asked reviewers to only rate papers on certain criteria and left the aggregation to meta reviewers. This approach can however lead to arbitrariness due to the differences in the aggregation approaches followed by different meta reviewers.

Noothigattu et al.[33] propose an algorithmic solution to this problem. They consider an often-suggested interface that asks reviewers to rate papers on a pre-specified set of criteria alongside their overall rating. Commensuration bias implies that each reviewer has their own subjective mapping of criteria to overall ratings. The key idea behind the proposed approach is to use machine learning and social choice theory to learn how the body of reviewers—at an aggregate level—map criteria to overall ratings. The algorithm then applies this learned mapping to the criteria ratings in each review to obtain a second set of overall ratings. The conference management system would then augment the reviewer-provided overall ratings with those computed using the learned mapping, with the primary benefit that the latter ratings are computed via the same mapping for all papers. This method was used in the AAAI 2022 conference to identify reviews with significant commensuration bias.

Confirmation bias. A controlled study by Mahoney[25] asked each reviewer to assess a fictitious manuscript. The contents of the manuscripts sent to different reviewers were identical in their reported experimental procedures but differed in their reported results. The study found that reviewers were strongly biased against papers with results that contradicted the reviewers' own prior views. The difference in the results section also manifested in other aspects: a manuscript whose results agreed with the reviewer's views was more likely to be rated as methodologically better, as having a better data presentation, and the reviewer was less likely to catch mistakes in the paper, even though these components were identical across the manuscripts.

Biases Regarding Author Identity

In 2015, two women researchers, Megan Head and Fiona Ingleby, submitted a paper to the PLOS ONE journal. A review they received read: *It would probably be beneficial to find one or two male researchers to work with (or at least obtain internal peer review from, but better yet as active co-authors*)." This is an example of how a review can take into consideration the authors' identities even when we expect it to focus exclusively on the scientific contribution.

Such biases with respect to author identities are widely debated in computer science and elsewhere. These debates have led to two types of peer-review processes: single-blind reviewing where reviewers are shown authors' identities, and double-blind reviewing where author identities are hidden from reviewers. In both settings, the reviewer identities are not revealed to authors.

A primary argument against single-blind reviewing is that it may cause the review to be biased with respect to the authors' identities. On the other hand, arguments against double-blind reviewing include: efforts to make a manuscript double blind, efficacy of double blinding (since many manuscripts are posted with author identities on preprint servers and social media), hindrance in checking (self-)plagiarism and conflicts of interest, and the use of author identities as a guarantee of trust for the details that reviewers have not been able to check carefully. In addition, the debate over single- vs-double-blind reviewing rests on the frequently asked question: "Where is the evidence of bias in single-blind reviewing in my field of research?"

A remarkable experiment was conducted at the Web Search and Data Mining (WSDM) 2017 conference,[45] which had 500 submitted papers and 1,987 reviewers. The reviewers were split randomly into two groups: a single-blind group and a double-blind group. Every paper was assigned two reviewers each from both groups. This experimental design allowed for a direct comparison of single-blind and double-blind reviews for each paper without requiring any additional reviewing for the experiment. The study found a significant bias in favor of famous authors, top universities, and top companies. Moreover, it found a non-negligible effect size but not statistically significant bias against papers with at least one woman author; the study also included a meta-analysis combining other studies, and this meta-analysis found this gender bias to be statistically significant. The study did not find evidence of bias with respect to papers from the U.S., nor when reviewers were from the same country as the authors, nor with respect to academic (versus industrial) institutions. The WSDM conference moved to double-blind reviewing the following year.

Another study[26] did not involve a controlled experiment but leveraged the fact that the ICLR conference switched from single blind to double blind reviewing in 2018. Analyzing both ratings and the text of reviews, the study found evidence of bias with respect to the affiliation of authors but not with respect to gender.

Such studies have also prompted a focus on careful design of experimental methods and measurement algorithms to evaluate biases in peer review, while mitigating confounding factors that may arise due to the complexity of the peer-review process.

Making reviewing double blind can mitigate these biases but may not fully eliminate them. Reviewers in three double-blind conferences were asked to guess the authors of the papers they were reviewing.[20] No author guesses were provided alongside 70%–86% of the reviews (it is not clear whether an absence of a guess indicates that the reviewer did not have a guess or if they did not wish to answer the question). However, among those reviews which did contain an author guess, 72%–85% guessed at least one author correctly.

In many research communities, it is common to upload papers to preprint servers such as arXiv (arxiv.org) before it is reviewed. For instance, 54% of all submissions to the NeurIPS 2019 conference were posted on arXiv and 21% of these submissions were seen by at least one reviewer. These preprints contain information about the authors, thereby potentially revealing the identities of the authors to reviewers. Based on these observations, one may be tempted to disallow authors from posting their manuscripts to preprint servers or elsewhere before they are accepted. However, one must tread this line carefully. First, such an embargo can hinder the progress of research. Second, the effectiveness of such prohibition is unclear. Studies have shown the content of the submitted paper can give clues about the identity of the authors.[20] Third, due to such factors, papers by famous authors may still be accepted at higher rates, while disadvantaged authors' papers neither get accepted nor can be put up on preprint servers.

These studies provide valuable quantitative information toward policy choices and trade-offs on blinded reviewing. That brings us to our next discussion on norms and policies.

The current research on improving peer review, particularly using computational methods, has only scratched the surface.

Norms and Policies

The norms and policies in any community or conference can affect the efficiency of peer review and the ability to achieve its goals.

Author incentives. Ensuring appropriate incentives for participants in peer review is a critical open problem: incentivizing reviewers to provide high-quality reviews and incentivizing authors to submit papers only when they are of suitably high quality.[2] We discuss some policies and associated effects pertaining to such author incentives, that are motivated by the rapid increase in the number of submissions in many conferences.

Open review. It is said that authors submitting a below-par paper have little to lose but lots to gain: hardly anyone will see the below-par version if it gets rejected, whereas the arbitrariness in the peer-review process gives it some chance of acceptance.

Some conferences are adopting an "open review" approach to peer review, where all submitted papers and their reviews (but not reviewer identities) are made public. A prominent example is the OpenReview.net conference management system in computer science. A survey[41] of participants at the ICLR 2013 conference, which was one of the first to adopt the open review format, pointed to increased accountability of authors as well as reviewers in this open format. An open reviewing approach also increases the transparency of the review process and provides more information to the public about the perceived merits/demerits of a paper rather than just a binary accept/reject decision.[2]

The open-review format can also result in some drawbacks; here is one such issue related to public visibility of rejected papers.

Resubmission policies. Many conferences are adopting policies where authors of a paper must provide past rejection information along with the submission. For instance, the 2020 International Joint Conference on Artificial Intelligence (IJCAI) required authors to prepend their submission with details of any previous rejections including prior reviews and the revisions made by authors. While these policies are well-intentioned toward ensuring that authors do not simply ignore reviewer feedback, the information of previous rejection could bias the reviewers.

A controlled experiment[43] tested for such a bias. Each reviewer was randomly shown one of two versions of a paper to review: one version indicated that the paper was previously rejected at another conference while the other version contained no such information. Reviewers gave almost one-point lower rating on a 10-point scale for the overall evaluation of a paper when they were told that a paper was a resubmission.

Rolling deadlines. In conferences with a fixed deadline, a large fraction of submissions are made on or very near the deadline. This observation suggests that removing deadlines (or in other words, having a "rolling deadline"), wherein a paper is reviewed whenever it is submitted, may allow authors ample time to write their paper as best as they can before submission, instead of cramming right before the fixed deadline. The flexibility offered by rolling deadlines may have additional benefits such as helping researchers better deal with personal constraints and allowing a more balanced sharing of resources such as compute.

The U.S. National Science Foundation experimented with this idea in certain programs.[15] The number of submitted proposals reduced drastically from 804 in one year in which there were two fixed deadlines, to just 327 in the subsequent 11 months when there was a rolling deadline. Thus, in addition to providing flexibility to authors, rolling deadlines may also help reduce the strain on the peer-review process.

Introduction to reviewing. While researchers are trained to do research, there is little training for peer review. Several initiatives and experiments have looked to address this challenge. Recently, the ICML 2020 conference adopted a method to select and then mentor junior reviewers, who would not have been asked to review otherwise, with a motivation of expanding the reviewer pool to address the large volume of submissions.[43] An analysis of their reviews revealed that the junior reviewers were more engaged through various stages of the process as compared to conventional reviewers. Moreover, the conference asked meta reviewers to rate all reviews, and 30% of reviews written by junior reviewers received the highest rating by meta reviewers, in contrast to 14% for the main pool.

Training reviewers at the beginning of their careers is a good start but may not be enough. There is some evidence[8] that quality of an individual's review falls over time, at a slow but steady rate, possibly because of increasing time constraints or in reaction to poor-quality reviews they themselves receive.

> **While researchers are trained to do research, there is little training for peer review ... Training reviewers at the beginning of their careers is a good start but may not be enough.**

Discussions and group dynamics. After submitting the initial reviews, reviewers of a paper are often allowed to see each other's' reviews. The reviewers and the meta reviewer then engage in a discussion to arrive at a final decision.

Several studies[34] conduct controlled experiments in the peer review of grant proposals to quantify the reliability of the process. The peer-review process studied here involves discussions among reviewers in panels. In each panel, reviewers first submit independent reviews, following which the panel engages in a discussion about the proposal, and reviewers can update their opinions. These studies reveal the following three findings. First, reviewers have quite a high level of disagreement with each other in their independent reviews. Second, the inter-reviewer disagreement within a panel decreases considerably after the discussions (possibly due to implicit or explicit pressure on reviewers to arrive at a consensus). This observation seems to suggest that the wisdom of all reviewers is being aggregated to make a more "accurate" decision. To quantify this aspect, these studies form multiple panels to evaluate each proposal, where each panel independently conducts the entire review process including the discussion. The studies then measure the amount of disagreement in the outcomes of the different panels for the same proposal. Their third finding is that, surprisingly, the level of disagreement across panels does *not* decrease after discussions, and instead often increases.

These observations indicate the need for a careful look at the efficacy of the discussion process and the protocols used therein. We discuss two experiments investigating potential reasons for the surprising reduction in the inter-panel agreement after discussions.

Teplitskiy et al.[44] conducted a controlled study to understand influence of other reviewers. They exposed reviewers to artificial ratings from other (fictitious) reviews. They found that 47% of the time, reviewers updated their ratings. Women reviewers updated their ratings 13% more frequently than men, and more so when they worked in male-dominated fields. Ratings that were initially high were updated downward 64% of the time, whereas ratings that were initially low were updated upward only 24% of the time.

Stelmakh et al.[43] investigated "herding" effects: Do discussions in peer review lead to the decisions getting biased toward the opinion of the reviewer who initiates the discussion? They found no evidence of such a bias.

Conclusion

The current research on improving peer review, particularly using computational methods, has only scratched the surface of this important application domain. There is much more to be done, with numerous open problems that are exciting and challenging, will be impactful when solved, and allow for an entire spectrum of theoretical, applied, and conceptual research.

Research on peer review faces at least two overarching challenges. First, there is no "ground truth" regarding which papers should have been accepted to the conference. One can evaluate individual modules of peer review and specific biases, as discussed in this article, but there is no well-defined measure of how a certain solution affected the entire process.

A second challenge is the unavailability of data. Research on improving peer review can significantly benefit from the availability of more data pertaining to peer review. However, a large part of the peer-review data is sensitive since the reviewer identities for each paper and other associated data are usually confidential. Designing policies and privacy-preserving computational tools to enable research on such data is an important open problem.

Nevertheless, there is increasing interest among research communities and conferences in improving peer review in a scientific manner. Researchers are conducting several experiments to understand issues and implications in peer review, designing methods and policies to address the various challenges, and translating research on this topic into practice. This bodes well for peer review, the cornerstone of scientific research. C

References

1. Alon, N., Fischer, F., Procaccia, A., and Tennenholtz, M. Sum of us: Strategy proof selection from the selectors. In *Proceedings of Conf. on Theoretical Aspects of Rationality and Knowledge*, (2011).
2. Anderson, T. Conference reviewing considered harmful. *ACM SIGOPS Operating Systems Rev.*, (2009).
3. Balietti, S., Goldstone, R., and Helbing, D. Peer review and competition in the art exhibition game. In *Proceedings of the National Academy of Sciences*, (2016).
4. Benos, D., et al. The ups and downs of peer review. *Advances in Physiology Education*, (2007).
5. Brenner, L., Griffin, D., and Koehler, D. Modeling patterns of probability calibration with random support theory: Diagnosing case-based judgment. *Organizational Behavior and Human Decision Processes*, (2005).
6. Brown, T. Peer review and the acceptance of new scientific ideas: Discussion paper from a working party on equipping the public with an understanding of peer peview: November 2002–May 2004. *Sense About Science*, (2004).
7. Cabanac, G. and Preuss, T. Capitalizing on order effects in the bids of peer-reviewed conferences to secure reviews by expert referees. *J. Assoc. Information Science and Tech.* (2013).
8. Callaham, M. and McCulloch, C. Longitudinal trends in the performance of scientific peer reviewers. *Annals of Emergency Medicine*, (2011).
9. Charlin, L. and Zemel, R. The Toronto Paper Matching System: An automated paper-reviewer assignment system. In *Proceedings of ICML Workshop on Peer Reviewing and Publishing Models*, (2013).
10. Fiez, T., Shah, N., and Ratliff, L. A SUPER* algorithm to optimize paper bidding in peer review. In *Proceedings of Conf. Uncertainty in Artificial Intelligence*, (2020).
11. Flach, P., Spiegler, S., Golénia, B., Price, S., Guiver, J., Herbrich, R., Graepel, T., and Zaki, M. Novel tools to streamline the conference review process: Experiences from SIGKDD'09. *SIGKDD Explor. Newsl*, (2010).
12. Freund, Y., Iyer, R., Schapire, R., and Singer, Y. An efficient boosting algorithm for combining preferences. *J. Machine Learning Research*, (2003).
13. Garg, N., Kavitha, T., Kumar, A., Mehlhorn, K., and Mestre, J. Assigning papers to referees. *Algorithmica*, (2010).
14. Guo, L., Wu, J., Chang, W., Wu, J., and Li, J. K-loop free assignment in conference review systems. In *Proceedings of ICNC*, (2018).
15. Hand, E. No pressure: NSF test finds eliminating deadlines halves number of grant proposals. *Science*, (2016).
16. Jecmen, S., Zhang, H., Liu, R., Shah, N., Conitzer, V., and Fang, F. Mitigating manipulation in peer review via randomized reviewer assignments. *NeurIPS*, (2020).
17. Jefferson, T., Alderson, P., Wager, E., and Davidoff, F. Effects of editorial peer review: a systematic review. *JAMA*, (2002).
18. Kobren, A., Saha, B., and McCallum, A. Paper matching with local fairness constraints. In *Proceedings of ACM KDD*, (2019).
19. Lauer, M. Case study in review integrity: Asking for favorable treatment. *NIH Extramural Nexus*, (2020).
20. Le Goues, C., Brun, Y., Apel, S., Berger, E., Khurshid, S., and Smaragdakis, Y. Effectiveness of anonymization in double-blind review. *Commun. ACM*, (2018).
21. Lee, C. Commensuration bias in peer review. *Philosophy of Science*, (2015).
22. Lee, C., Sugimoto, C., Zhang, G., and Cronin, B. Bias in peer review. *J. Assoc. Information Science and Technology*, (2013).
23. Leyton-Brown, K. and Mausam. AAAI 2021—Introduction; https://bit.ly/3r2L3Rr; (min. 8 onward).
24. Littman, M. Collusion rings threaten the integrity of computer science research. *Commun. ACM*, (2021).
25. Mahoney, M. Publication prejudices: an experimental study of confirmatory bias in the peer review system. *Cognitive Therapy and Research* (1977).
26. Manzoor, E. and Shah, N. Uncovering latent biases in text: Method and application to peer review. In *Proceedings of AAAI*, (2021).
27. Markwood, I., Shen, D., Liu, Y., and Lu, Z. Mirage: Content masking attack against information-based online services. In *Proceedings of USENIX Security Symp.*, (2017).
28. Mattei, N. and Walsh, T. Preflib: A library for preferences. In *Proceedings of Intern. Conf. Algorithmic Decision Theory*. Springer, 2013; http://www.preflib.org
29. Mimno, D. and McCallum, A. Expertise modeling for matching papers with reviewers. In *Proceedings of KDD*, (2007).
30. Mitliagkas, I., Gopalan, A., Caramanis, C., and Vishwanath, S. User rankings from comparisons: Learning permutations in high dimensions. In *Proceedings of Allerton Conf.*, (2011).
31. Murphy, J., Hofacker, C., and Mizerski, R. Primacy and recency effects on clicking behavior. *J. Computer-Mediated Commun.*, (2006).
32. Neubig, G., Wieting, J., McCarthy, A., Stent, A., Schluter, N., and Cohn, T. ACL reviewer matching code; https://github.com/acl-org/reviewerpaper-matching.
33. Noothigattu, R., Shah, N., and Procaccia, A. Loss functions, axioms, and peer review. *J. Artificial Intelligence Research*, (2021).
34. Pier, E., Raclaw, J., Kaatz, A., Brauer, M., Carnes, M., Nathan, M., and Ford, C. Your comments are meaner than your score: Score calibration talk influences intra-and inter-panel variability during scientific grant peer review. *Research Evaluation* (2017).
35. Rennie, D. Let's make peer review scientific. *Nature*, (2016).
36. Rodriguez, M., Bollen, J., and Van de Sompel, H. Mapping the bid behavior of conference referees. *J. Informetrics* (2007).
37. Roos, M., Rothe, J., and Scheuermann, B. How to calibrate the scores of biased reviewers by quadratic programming. In *Proceedings of AAAI*, (2011).
38. Shah, N., Tabibian, B., Muandet, K., Guyon, I., and Von Luxburg, U. Design and analysis of the NIPS 2016 review process. *JMLR*, (2018).
39. Siegelman, S. Assassins and zealots: Variations in peer review. *Radiology*, (1991).
40. Smith, R. Peer review: Reform or revolution? Time to open up the black box of peer review. (1997).
41. Soergel, D., Saunders, A., and McCallum, A. Open scholarship and peer review: A time for experimentation, (2013).
42. Stelmakh, I., Shah, N., and Singh, A. PeerReview4All: Fair and accurate reviewer assignment in peer review. *JMLR*, (2021).
43. Stelmakh, I., Shah, N., Singh, A., Daumé III, H., and Rastogi, C. Experiments with the ICML 2020 peer-review process, (2020); https://blog.ml.cmu.edu/2020/12/01/icml2020exp/
44. Teplitskiy, M., Ranub, H., Grayb, G., Meniettid, M., Guinan, E., and Lakhani, K. Social influence among experts: Field experimental evidence from peer review, (2019).
45. Tomkins, A., Zhang, M., and Heavlin, W. Reviewer bias in single-versus double-blind peer review. In *Proceedings of the National Academy of Sciences*, (2017).
46. Vijaykumar, T. Potential organized fraud in ACM/IEEE computer architecture conferences, (2020); https://bit.ly/3o2Zjb3.
47. Wang, J. and Shah, N. Your 2 is my 1, your 3 is my 9: Handling arbitrary miscalibrations in ratings. In *Proceedings of AAMAS*, (2019).
48. Ware, M. Publishing research consortium peer review survey 2015. *Publishing Research Consortium*, (2016).
49. Wing, J. and Chi, E. Reviewing peer review. *Communications of the ACM* (2011).
50. Wu, R., Guo, C., Wu, F., Kidambi, R., van der Maaten, L., and Weinberger, K. Making paper reviewing robust to bid manipulation attacks, (2021); *arXiv:2102.06020*.
51. Xu, Y., Zhao, H., Shi, X., and Shah, N. On strategyproof conference review. In *Proceedings of IJCAI*, (2019).

An extended version of this article discussing additional challenges, experiments, and solutions is available at http://bit.ly/PeerReviewOverview.

Nihar B. Shah is an assistant professor in the Machine Learning and Computer Science departments of Carnegie Mellon University, Pittsburgh, PA, USA.

DOI:10.1145/3531443

To view the accompanying paper, visit doi.acm.org/10.1145/3531445

Technical Perspective
The Compression Power of the BWT

By Gonzalo Navarro

MASSIVE AND HIGHLY repetitive text collections are arising in several modern applications. For example, a U.K. project managed in 2018 to sequence 100,000 human genomes, which stored in plain form require 300 terabytes. Further, the data structures needed to efficiently perform the complex searches required in bioinformatics would add another order of magnitude to the storage space, reaching the petabytes.

How to cope with this flood of repetitive data? We can think of compression (after all, two human genomes differ by about 0.1%), but it is not the definitive answer—we need a way to decompress the data before we can use it. A more ambitious research area, *compressed data structures*, promises to store the data *and* the structures required to efficiently handle it, within space close to that of the compressed data. The data will never be decompressed; it will always be used directly in compressed form.

However, on these repetitive datasets, statistical compression is useless. Dictionary compression techniques like Lempel-Ziv perform much better, because they replace text chunks by references to identical chunks seen before. Lempel-Ziv can compress our genome collections by a factor of 100, to three affordable terabytes. But again, this is just compression. Can we design compressed data structures based on dictionary compression, that extract snippets, and even search the genomes efficiently, without decompressing?

There has been a good deal of research on this challenge. Even extracting a snippet without decompressing the text from the beginning is tricky. We don't know how to do it on Lempel-Ziv, but it is possible on slightly weaker dictionary compression formats, like grammar compression (where one finds a small context-free grammar that generates the text). Providing efficient searches is even more difficult: on those huge collections, sequentially scanning the text is not a choice. We must build an *index* data structure that accelerates searches. Unlike in statistical compression, dictionary compressors tend to cut the text into pieces, so a substring sought can appear in many different forms. Despite those challenges, there are currently various compressed indexes that provide efficient access and search for short strings, and whose size is bounded in terms of the size z of the Lempel-Ziv encoding or the size g of grammar compression.

Bioinformatic applications require more complex searches. They need to search allowing errors, to search for all the substrings of a long string, to find frequent long enough text substrings, to find the longest substrings that appear repeated in the text, and many others.

A beloved data structure in stringology, the *suffix tree*, can efficiently answer all those complex queries. However, it requires a lot of space—an order of magnitude over the plain text. To be used in bioinformatics, it underwent various simplifications and space reductions. Researchers showed how to do it with just *suffix arrays*—the leaves of the *suffix tree*—and then with the *FM-index*—a statistically compressed suffix array.

The FM-index builds on the Burrows-Wheeler Transform (BWT), a permutation of the text that makes it easier to compress. It was soon discovered the BWT featured runs of equal consecutive symbols that became longer as the text was more repetitive. The number r of runs in the BWT became then a measure of repetitiveness—just like z or g. Further research managed, within space bounded in terms of r, not only to extract snippets and perform basic searches, but also to support all the complex searches offered by suffix trees (this compression does not cut the text, so things are simpler).

The following paper is a beautiful masterpiece … written by two rising stars in combinatorial pattern matching.

In parallel, researchers aimed to understand the compressibility limits of repetitive texts, obtaining lower bounds in terms of the number of different substrings. Over time, it has turned out that all the compressibility measures and lower bounds are sandwiched within a logarithmic factor of each other, so they are all relatively close—except r.

The measure r, which offered a world of sophisticated searches, seemed to be an outlier. It sat between statistical and dictionary compression and was the only measure that could not be bounded in terms of the others. In practice, the structures based on r were indeed larger than those based on z or g, which raised concerns on how well r captured repetitiveness.

The following paper finally settles the question. It proves that r is at most a log-square factor away from z, which validates r as a measure of repetitiveness. At the same time, it confirms previous intuition that r is not that small. But at least we now know the price for going from basic to complex string searches—what is needed in bioinformatics.

Besides the relevance of the result, the paper is a beautiful masterpiece and explores many other consequences. It is written by two rising stars in combinatorial pattern matching. I hope to hear much more from them in the years to come.

Gonzalo Navarro is a professor in the Department of Computer Science at the University of Chile.

DOI:10.1145/3531445

To view the accompanying Technical Perspective, visit doi.acm.org/10.1145/3531443

Resolution of the Burrows-Wheeler Transform Conjecture

By Dominik Kempa and Tomasz Kociumaka

Abstract

The Burrows-Wheeler Transform (BWT) is an invertible text transformation that permutes symbols of a text according to the lexicographical order of its suffixes. BWT is the main component of popular lossless compression programs (such as `bzip2`) as well as recent powerful compressed indexes (such as the r-index[7]), central in modern bioinformatics. The compressibility of BWT is quantified by the number r of equal-letter runs in the output. Despite the practical significance of BWT, no nontrivial upper bound on r is known. By contrast, the sizes of nearly all other known compression methods have been shown to be either always within a polylog n factor (where n is the length of the text) from z, the size of Lempel–Ziv (LZ77) parsing of the text, or much larger in the worst case (by an n^ε factor for $\varepsilon > 0$).

In this paper, we show that $r = \mathcal{O}(z \log^2 n)$ holds for every text. This result has numerous implications for text indexing and data compression; in particular: (1) it proves that many results related to BWT automatically apply to methods based on LZ77, for example, it is possible to obtain functionality of the suffix tree in $\mathcal{O}(z \text{ polylog } n)$ space; (2) it shows that many text processing tasks can be solved in the optimal time assuming the text is compressible using LZ77 by a sufficiently large polylog n factor; and (3) it implies the first nontrivial relation between the number of runs in the BWT of the text and of its reverse.

In addition, we provide an $\mathcal{O}(z \text{ polylog } n)$-time algorithm converting the LZ77 parsing into the run-length compressed BWT. To achieve this, we develop several new data structures and techniques of independent interest. In particular, we define compressed string synchronizing sets (generalizing the recently introduced powerful technique of string synchronizing sets[11]) and show how to efficiently construct them. Next, we propose a new variant of wavelet trees for sequences of long strings, establish a nontrivial bound on their size, and describe efficient construction algorithms. Finally, we develop new indexes that can be constructed directly from the LZ77 parsing and efficiently support pattern matching queries on text substrings.

1. INTRODUCTION

Lossless data compression aims to exploit redundancy in the input data to represent it in a small space. Despite the abundance of compression methods, nearly every existing tool falls into one of the few general frameworks, among which the three most popular are the following: Lempel–Ziv compression (where the nominal and most commonly used is the LZ77 variant[25]), statistical compression (this includes, e.g., context mixing, prediction by partial matching (PPM), and dynamic Markov coding), and Burrows-Wheeler transform (BWT).[4]

One of the features that best differentiates these algorithms is whether they better remove the redundancy caused by skewed symbol frequencies or by repeated fragments. The idea in LZ77 (which underlies, e.g., `7-zip` and `gzip` compressors) is to partition the input text into long substrings, each having an earlier occurrence in the text. Every substring is then encoded as a pointer to the previous occurrence using a pair of integers. This method natively handles long repeated substrings and can achieve an exponential compression ratio given sufficiently repetitive input. Statistical compressors, on the other hand, are based on representing (predicting) symbols in the input based on their frequencies. This is formally captured by the notion of the *kth order empirical entropy* $H_k(T)$. For any sufficiently long text T, symbol frequencies (taking length-k contexts into account) in any power of T (the concatenation of several copies of T) do not change significantly (see Kreft and Navarro,[14] Lemma 2.6). Therefore, $|T^t|\, H_k(T^t) \approx t \cdot |T| H_k(T)$ holds for any $t \geq 1$, which means that entropy is not sensitive to long repetitions, and hence it is not able to capture the same type of redundancy as the LZ77 compression.[7,12,14,24]

The above analysis raises the question about the nature of compressibility of the Burrows-Wheeler transform. The compression of BWT-based compressors, such as `bzip2`, is quantified by the number r of equal-letter runs in the BWT. The clear picture described above no longer applies to the measure r. On the one hand, Manzini[16] proved that r can be upper-bounded in terms of the kth order empirical entropy of the input string. On the other hand, already in 2008, Sirén et al.[24] observed that BWT achieves excellent compression (superior to statistical methods) on highly repetitive collections and provided probabilistic analysis exhibiting cases when r is small. Yet, after more than a decade, no upper bound on r in terms of z (the size of the LZ77 parsing) was discovered.

This lack of understanding is particularly frustrating due to numerous applications of BWT in the field of bioinformatics and compressed computation. One of the most

The original version of this paper was presented at FOCS 2020 and is available at https://arxiv.org/abs/1910.10631

successful applications of BWT is in *compressed indexing*, which aims to store a compressed string simultaneously supporting various queries (such as random access, pattern matching, or even suffix array queries) concerning the uncompressed version. Although classical (uncompressed) indexes, such as suffix trees and suffix arrays, have been successful in many applications, they are not suitable for storing and searching big highly repetitive collections, which are virtually impossible to search without preprocessing. A recent survey[17] provides several real-life examples of such datasets. In particular, Github stores more than 20 terabytes of data, with an average of over 20 versions per project, whereas the 100000 Human Genome Project produced over 70 terabytes of DNA, which is highly compressible due to 99.9% similarity between individual human genomes. Motivated by such applications, compressed indexing witnessed a remarkable surge of interest in recent years. BWT-based indexes, such as the r-index,[7] are among the most powerful, and their space usage is up to $\mathcal{O}(\text{polylog } n)$ factors away from the value r. For a comprehensive overview of this field, we refer the reader to a survey by Navarro.[18]

In addition to text indexing, the Burrows-Wheeler transform has many applications in computational biology. In particular, BWT is the main component of the popular genome read aligners such as `Bowtie`, `BWA`, and `Soap2`, which paved its way to general bioinformatics textbooks.[21] Specialized literature on algorithmic analysis of biological sequences[15,19] devotes dozens of pages to BWT applications.

Given the importance and practical significance of BWT, one of the biggest open problems that emerged in the field of lossless data compression and compressed computation asks:

What is the upper bound on the output size of the Burrows-Wheeler transform?

Except for BWT, essentially every other known compression method has been proven to produce output whose size is always within an $\mathcal{O}(\text{polylog } n)$ factor from z, the output size of the LZ77 algorithm (e.g., grammar compression, collage systems, and macro schemes)[a] or larger by a polynomial factor (n^ε for some $\varepsilon > 0$) in the worst case (e.g., LZ78, compressed word graphs (CDAWGs)). We refer the reader to Navarro[17] for a survey of repetitiveness measures. Notably, BWT is known to never compress much better than LZ77, that is, $z = \mathcal{O}(r \log n)$,[6] and the opposite relation $r = \mathcal{O}(z \text{ polylog } n)$ was often conjectured to be false.[5]

1.1. Our contribution

We prove that $r = \mathcal{O}(z \log^2 n)$ holds for all strings, resolving the BWT conjecture in the more surprising way and answering an open question of Gagie et al.[5,6] This result alone has multiple implications for indexing and compression:

1. It is possible to support suffix array and suffix tree functionality in $\mathcal{O}(z \text{ polylog } n)$ space.[7]
2. It was shown by Kempa[10] that many string processing tasks (such as BWT and LZ77 construction) can be solved in $\mathcal{O}(n/\log_\sigma n + r \text{ polylog } n)$ time (where σ is the alphabet size). Thus, if the text is sufficiently compressible by BWT (formally, $n/r = \Omega$ (polylog n)), these tasks can be solved in optimal time (which is unlikely to be possible for general texts[11]). Our result loosens this assumption to $n/z = \Omega$ (polylog n).
3. Until now, methods based on the Burrows-Wheeler transform were thought to be neither statistical nor dictionary (LZ-like) compression algorithms.[5,24] Our result challenges the notion that the BWT forms a separate compression type: Because of our bound, BWT is much closer to LZ compressors than anticipated.

Our slightly stronger bound $r = \mathcal{O}(\delta \log^2 n)$, where $\delta \le z$ is a symmetric (insensitive to string reversal) repetitiveness measure recently studied by Kociumaka et al.,[13] further shows that:

4. The number $\overline{r}$ of BWT runs in the reverse of the text satisfies $\overline{r} = \mathcal{O}(r \log^2 n)$, which is the first nontrivial bound in terms of r. This result is of practical importance due to many algorithms whose efficiency depends on $\overline{r}$.[2,20,22,23]

After proving $r = \mathcal{O}(z \log^2 n)$, we refine our approach to obtain $r = \mathcal{O}(z \log z \max(1, \log \frac{n}{z \log z}))$ and subsequently $r = \mathcal{O}(\delta \log \delta \max(1, \log \frac{n}{\delta \log \delta}))$. We then show that the latter bound, combined with the trivial one $r \le n$, is asymptotically tight for the full spectrum of values of n and δ. As a side result, we obtain a tight upper bound $\mathcal{O}(n \log \delta)$ on the sum of irreducible LCP values, improving upon the previously known bound $\mathcal{O}(n \log r)$.[9]

Next, we develop an $\mathcal{O}(z \log^8 n)$-time algorithm converting the LZ77 parsing into the run-length compressed BWT (the polylog n factor has not been optimized). This offers up to exponential speedup over the previously fastest space-efficient algorithms, which need $\Omega(n \log z)$ time.[20,22] To achieve this, we develop new data structures and techniques of independent interest. In particular, we introduce a notion of compressed string synchronizing sets, generalizing the technique by Kempa and Kociumaka.[11] We then describe a new variant of wavelet trees,[8] designed to work for sequences of long strings. In the full version of this paper, we also propose new indexes that can be built directly from the LZ77 parsing and support fast pattern matching queries on text substrings.

2. PRELIMINARIES

A *string* is a finite sequence of characters from a given *alphabet*. The length of a string S is denoted by $|S|$ and, for $i \in [1 \mathinner{.\,.} |S|]$,[b] the ith character of S is denoted by $S[i]$. A string U is a *substring* of S if $U = S[i]\,S[i+1] \ldots S[j-1]$ for some $1 \le i \le j \le |S| + 1$. We then say that U *occurs* in S at position i. The *occurrence* of U at position i in S is denoted by $S[i \mathinner{.\,.} j)$ or $S[i \mathinner{.\,.} j-1]$. Such an occurrence is a *fragment* of S and can

a The choice for LZ77 as a representative in this class follows from the fact that most other methods are NP-hard to optimize, whereas LZ77 admits a simple linear-time compression.

b For integers $i, j \in \mathbb{Z}$, we denote $[i \mathinner{.\,.} j] = \{k \in \mathbb{Z} : i \le k \le j\}$, $[i \mathinner{.\,.} j) = \{k \in \mathbb{Z} : i \le k < j\}$, and $(i \mathinner{.\,.} j] = \{k \in \mathbb{Z} : i < k < j\}$.

be represented by (a pointer to) S and the two positions i, j. Two fragments (perhaps of different strings) *match* if they are occurrences of the same substring. A fragment $S[i\,..\,j]$ is a *prefix* if $i = 1$ and a *suffix* if $j = |S|$.

We use $\overline{S}$ to denote the *reverse* of S, that is, $S[|S|] \ldots S[2]\,S[1]$. We denote the *concatenation* of two strings U and V, that is, $U[1]U[2] \ldots U[|U|]V[1]V[2] \ldots V[|V|]$, by UV or $U \cdot V$. Furthermore, $S^k := \bigodot_{i=1}^{k} S$ denotes the concatenation of $k \geq 0$ copies of S; note that $S^0 = \varepsilon$ is the *empty string*.

An integer $p \in [1\,..\,|S|]$ is a *period* of a string S if $S[i] = S[i + p]$ holds for every $i \in [1\,..\,|S| - p]$. The shortest period of S is denoted as per(S). A string S is called *periodic* if $\operatorname{per}(S) \leq \frac{1}{2}|S|$. The following fact is a consequence of the classic *periodicity lemma* (which we do not use directly).

Fact 2.1 (see Amir et al.,[1] Fact 1). *Any two distinct periodic strings of the same length differ on at least two positions.*

Throughout the paper, we consider a string (called the *text*) T of length $n \geq 1$ over an ordered alphabet Σ of size σ. We assume that $T[n] = \$$, where \$ is the smallest symbol in Σ, and that \$ does not occur anywhere else in T. We use $\preceq$ to denote the order on Σ, extended to the *lexicographic* order on Σ^* (the set of strings over Σ) so that $U, V \in \Sigma^*$ satisfy $U \preceq V$ if and only if either U is a prefix of V, or $U[1\,..\,i) = V[1\,..\,i)$ and $U[i] \prec V[i]$ holds for some $i \in [1\,..\,\min(|U|, |V|)]$.

The *suffix array* SA$[1\,..\,n]$ of T is a permutation of $[1\,..\,n]$ such that $T[\mathrm{SA}[1]\,..\,n] \prec T[\mathrm{SA}[2]\,..\,n] \prec \ldots \prec T[\mathrm{SA}[n]\,..\,n]$. The closely related *Burrows-Wheeler transform* BWT$[1\,..\,n]$ of T is defined by $\mathrm{BWT}[i] = T[\mathrm{SA}[i] - 1]$ if $\mathrm{SA}[i] > 1$ and $\mathrm{BWT}[i] = T[n]$ otherwise. In the context of BWT, it is convenient to consider an *infinite string* T^∞ defined so that $T^\infty[i] = T[1 + (i - 1) \bmod n]$ for $i \in \mathbb{Z}$; in particular, $T^\infty[1\,..\,n] = T[1\,..\,n]$. We then have $\mathrm{BWT}[i] = T^\infty[\mathrm{SA}[i] - 1]$ for $i \in [1\,..\,n]$. Moreover, the assumption $T[n] = \$$ implies $T^\infty[\mathrm{SA}[1]\,..\,] \prec \ldots \prec T^\infty[\mathrm{SA}[n]\,..\,]$.

For any string $S = c_1^{\ell_1} c_2^{\ell_2} \cdots c_h^{\ell_h}$, where $c_i \in \Sigma$ and $\ell_i \in \mathbb{Z}_{>0}$ for $i \in [1\,..\,h]$, and $c_i \neq c_{i+1}$ for $i \in [1\,..\,h)$, we define the *run-length encoding* of S as a sequence $\mathrm{RL}(S) = ((c_1, \lambda_1), \ldots, (c_h, \lambda_h))$, where $\lambda_i = \ell_1 + \ldots + \ell_i$ for $i \in [1\,..\,h]$. Throughout, we let $r = |\mathrm{RL}(\mathrm{BWT})|$ denote the number of runs in the BWT of T. For example, for the text $T =$ `bbabaababababaababa$` in Table 1, we have BWT $=$ a^{1}b^{6}a^{1}b^{2}a^{6}b^{1}a^2\$^{1}, and hence $r = 8$.

By lcp(U, V) we denote the length of the longest common prefix of strings U and V. For $j_1, j_2 \in [1\,..\,n]$, we let $\mathrm{LCE}(j_1, j_2) = \mathrm{lcp}(T[j_1\,..\,], T[j_2\,..\,])$. The *LCP array* LCP$[1\,..\,n]$ is defined so that $\mathrm{LCP}[i] = \mathrm{LCE}(\mathrm{SA}[i], \mathrm{SA}[i-1])$ for $i \in [2\,..\,n]$ and $\mathrm{LCP}[1] = 0$. We say that the value LCP[i] is *reducible* if $\mathrm{BWT}[i] = \mathrm{BWT}[i-1]$ and *irreducible* otherwise (including when $i = 1$). There are exactly r irreducible LCP values.

We say that a nonempty fragment $T[i\,..\,i{+}\ell)$ is a *previous factor* if it has an earlier occurrence in T, that is, $\mathrm{LCE}(i, i') \geq \ell$ holds for some $i' \in [1\,..\,i)$. The *LZ77 parsing* of T is a factorization $T = F_1 \cdots F_f$ into nonempty *phrases* such that the jth phrase F_j is the longest previous factor starting at position $1 + |F_1 \cdots F_{j-1}|$; if no previous factor starts there, then F_j consists of a single character. In the underlying *LZ77 representation,* every phrase $F_j = T[i\,..\,i{+}\ell)$ that is a previous factor is encoded as (i', ℓ), where $i' \in [1\,..\,i)$ satisfies $\mathrm{LCE}(i, i') \geq \ell$ (and is chosen arbitrarily in case of multiple possibilities); if $F_j = T[i]$ is not a previous factor, then it is encoded as $(T[i], 0)$. We denote the number of phrases in the LZ77 parsing by z. For example, the text `bbabaababababaababa$` of Table 1 has LZ77 factorization `b`·`b`·`a`·`ba`·`aba`·`bababa`·`ababa`·`$` with $z = 8$ phrases, and its LZ77 representation is (b, 0), (1, 1), (a, 0), (2, 2), (3, 3), (7, 6), (10, 5), (\$, 0).

Table 1. The lexicographically sorted suffixes of a string $T =$ `bbabaababababaababa$` along with the BWT, SA, and LCP tables.

i	SA[i]	LCP[i]	BWT[i]	T[SA[i] .. n]
1	20	**0**	a	`$`
2	19	**0**	b	`a$`
3	14	1	b	`aababa$`
4	5	6	b	`aababababaababa$`
5	17	1	b	`aba$`
6	12	3	b	`abaababa$`
7	3	8	b	`abaababababaababa$`
8	15	**3**	a	`ababa$`
9	10	**5**	b	`ababaababa$`
10	8	5	b	`abababaababa$`
11	6	**7**	a	`ababababaababa$`
12	18	0	a	`ba$`
13	13	2	a	`baababa$`
14	4	7	a	`baababababaababa$`
15	16	2	a	`baba$`
16	11	4	a	`babaababa$`
17	2	**9**	b	`babaababababaababa$`
18	9	**4**	a	`bababaababa$`
19	7	6	a	`babababaababa$`
20	1	**1**	\$	`bbabaababababaababa$`

The irreducible LCP values are bold and underlined.

The *trie* of a finite set $\mathcal{S} \subseteq \Sigma^*$ is an edge-labeled rooted tree with a node v_X for every string X that is a prefix of a string $S \in \mathcal{S}$. The trie is rooted at v_ε and the parent of each node v_X with $X \neq \varepsilon$ is $v_{X[1\,..\,|X|)}$. In this case, the edge from $v_{X[1\,..\,|X|)}$ to v_X is *labeled* with $X[|X|]$. See Figure 1 for the trie of {`$aba`, `aaba`, `abaa`, `abab`, `abb$`, `b$ab`, `baab`, `baba`, `babb`, `bb$a`}.

3. COMBINATORIAL BOUNDS

3.1. Basic upper bound

To illustrate the main idea of our proof technique, we first prove the upper bound in its simplest form $r = \mathcal{O}(z \log^2 n)$. The following lemma stands at the heart of our proof.

Lemma 3.1. *For every $\ell \in [1\,..\,n]$, the number of irreducible LCP values in $[\ell\,..\,2\ell)$ is $\mathcal{O}(z \log n)$.*

Proof. Denote $\mathcal{S}_m = \{S \in \Sigma^m : S \text{ is a substring of } T^\infty\}$ for $m \geq 1$. Observe that $|\mathcal{S}_m| \leq mz$ because every length-m substring of T^∞ has an occurrence crossing or beginning at a phrase boundary of the LZ77 parsing of T. This includes substrings overlapping two copies of T, which cross the boundary between the last and the first phrase.

The idea of the proof is as follows: With each irreducible value $\mathrm{LCP}[i] \in [\ell\,..\,2\ell)$, we associate a cost of ℓ units, which are charged to individual characters of strings in $\mathcal{S}_{3\ell}$. We then show that each of the strings in $\mathcal{S}_{3\ell}$ is charged at most 2 $\log n$ times. The number of irreducible LCP values in $[\ell\,..\,2\ell)$ equals $\frac{1}{\ell}$ times the total cost, which is at most

$$|\mathcal{S}_{3\ell}| \cdot 2 \log n \leq 6\ell z \log n.$$

Figure 1. The trie $\mathcal{T}$ of the reversed length-4 substrings of T^∞ for T = `bbabaababababaababa$` of Table 1. Light edges are thin and dotted.

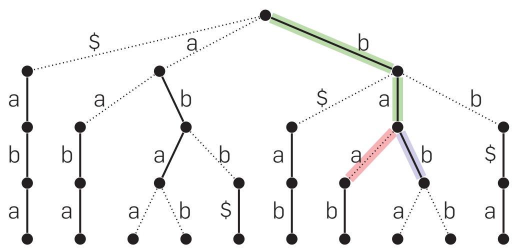
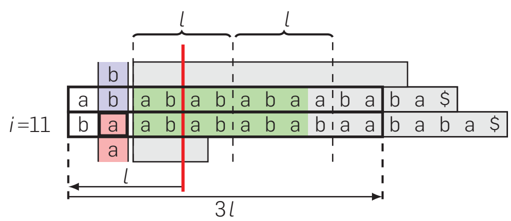


Figure 2. Proof of Lemma 3.1 on T from Table 1 for $\ell = 4$, $i = 11$, and $k = 2$. Strings $T^\infty[j_t - k\,..\,j_t - k + 3\ell)$ are highlighted. The subtree of $\mathcal{T}$ rooted in ν_{baa} is smaller than in ν_{bab} (see Figure 1), and hence we charge the second symbol of $T^\infty[j_1 - k\,..\,j_1 - k + 3\ell)$, that is, $t = 1$.



To devise the announced assignment of cost to characters of strings in $\mathcal{S}_{3\ell}$, consider the trie $\mathcal{T}$ of all reversed strings in $\mathcal{S}_\ell$ (see Figure 1 for example).

Let $\mathrm{LCP}[i] \in [\ell\,..\,2\ell)$ be an irreducible LCP value; note that $i > 1$ due to $\mathrm{LCP}[i] \geq \ell > 0$. Let $j_0 = \mathrm{SA}[i-1]$ and $j_1 = \mathrm{SA}[i]$ so that $\mathrm{LCP}[i] = \mathrm{LCE}(j_0, j_1)$. Because $\mathrm{LCP}[i]$ is irreducible, we have $T^\infty[j_0 - 1] = \mathrm{BWT}[i-1] \neq \mathrm{BWT}[i] = T^\infty[j_1 - 1]$. For $k \in [1\,..\,\ell]$, the kth unit of the cost associated with $\mathrm{LCP}[i]$ is charged to the kth character $(T^\infty[j_t - 1])$ of the string $T^\infty[j_t - k\,..\,j_t - k + 3\ell) \in \mathcal{S}_{3\ell}$, where $t \in \{0, 1\}$ is such that the subtree of $\mathcal{T}$ rooted at $\nu_{\overline{T^\infty[j_t - 1\,..\,j_t - k + \ell)}}$ contains fewer leaves than the subtree rooted at $\nu_{\overline{T^\infty[j_{1-t} - 1\,..\,j_{1-t} - k + \ell)}}$ (we choose $t = 0$ in case of ties); see Figure 2 for an illustration.

Note that at most $\log n$ characters of each $S \in \mathcal{S}_{3\ell}$ can be charged during the above procedure: Whenever $S[k]$, with $k \in [1\,..\,\ell]$, is charged, the subtree of $\mathcal{T}$ rooted at $\nu_{\overline{S[k+1\,..\,\ell]}}$ has at least twice as many leaves as the subtree rooted at $\nu_{\overline{S[k\,..\,\ell]}}$, and this can happen for at most $\log|\mathcal{S}_\ell| \leq \log n$ nodes $\nu_{\overline{S[k\,..\,\ell]}}$ on the path from $\nu_{\overline{S[1\,..\,\ell]}}$ to the root of $\mathcal{T}$.

It remains to show that, for every $S \in \mathcal{S}_{3\ell}$, a single position $S[k]$, with $k \in [1\,..\,\ell]$, can be charged at most twice. For this, observe that the characters charged for a single irreducible value $\mathrm{LCP}[i]$ are at different positions (of strings in $\mathcal{S}_{3\ell}$). Hence, to analyze the total charge assigned to $S[k]$, we only need to bound the number of possible candidate positions i. Let $[b\,..\,e]$ be the set of indices i' such that $T^\infty[\mathrm{SA}[i']\,..)$ starts with $S[k+1\,..\,3\ell]$. In the above procedure, if a character $S[k]$ is charged a unit of cost corresponding to $\mathrm{LCP}[i]$, then $S[k+1\,..\,3\ell]$ is a prefix of either $T^\infty[\mathrm{SA}[i-1]\,..) = T^\infty[j_0\,..)$ or $T^\infty[\mathrm{SA}[i]\,..) = T^\infty[j_1\,..)$. Hence, $\{i-1, i\} \cap [b\,..\,e] \neq \emptyset$. At the same time, $\mathrm{LCE}(\mathrm{SA}[i-1], \mathrm{SA}[i]) < 2\ell$, and all strings $T^\infty[\mathrm{SA}[i']\,..)$ with $i' \in [b\,..\,e]$ share a common prefix $S[k+1\,..\,3\ell]$ of length $3\ell - k \geq 2\ell$. Thus, $i = b$ or $i = e + 1$. □

Theorem 3.2. *All length-n strings satisfy $r = \mathcal{O}(z \log^2 n)$.*

Proof. Recall that r is the total number of irreducible LCP values. Thus, the claim follows by applying Lemma 3.1 for $\ell_i = 2^i$, with $i \in [0\,..\lfloor \log n \rfloor]$, and observing that the number of LCP values 0 is exactly $\sigma \leq z$. □

3.2. Tighter upper bound

To obtain a tighter bound, we refine the ideas from Section 3.1, starting with a counterpart of Lemma 3.1.

Lemma 3.3. *For every $\ell \in [1\,..\,n]$, the number of irreducible LCP values in $[\ell\,..\,2\ell)$ is $\mathcal{O}(z \log z)$.*

Proof. The proof follows closely that of Lemma 3.1. However, with each irreducible value $\mathrm{LCP}[i] \in [\ell\,..\,2\ell)$, we associate cost $\lceil \frac{1}{2}\ell \rceil$ instead of ℓ. We then show that each of the strings in $\mathcal{S}_{3\ell}$ is charged at most $2 \cdot (3 + \log z)$ times (rather than $2 \log n$ times). Then, the number of irreducible LCP values in the range $[\ell\,..\,2\ell)$ does not exceed $\frac{2}{\ell}$ times the total cost, which is bounded by

$$|\mathcal{S}_{3\ell}| \cdot 2 \cdot (3 + \log z) \leq 6\ell z(3 + \log z).$$

Recall the trie $\mathcal{T}$ of all reversed strings in $\mathcal{S}_\ell$. For a node ν of $\mathcal{T}$, by $\mathrm{size}(\nu)$, we denote the number of leaves in the subtree of $\mathcal{T}$ rooted in ν. An edge connecting $\nu \neq \mathrm{root}(\mathcal{T})$ to its parent in $\mathcal{T}$ is called *light* if ν has a sibling ν' satisfying $\mathrm{size}(\nu') \geq \mathrm{size}(\nu)$ (see Figure 1). In the proof of Lemma 3.1, we observed that the characters $S[k]$ of $S \in \mathcal{S}_{3\ell}$ that can be charged correspond to light edges on the path from the root of $\mathcal{T}$ to the leaf $\nu_{\overline{S[1\,..\,\ell]}}$: Whenever $S[k]$, with $k \in [1\,..\,\ell]$, is charged, the edge connecting $\nu_{\overline{S[k\,..\,\ell]}}$ to its parent $\nu_{\overline{S[k+1\,..\,\ell]}}$ is light. We then noted that there are at most $\log|\mathcal{S}_\ell| \leq \log n$ light edges on each root-to-leaf path in $\mathcal{T}$. Here, we charge the characters of strings in $\mathcal{S}_{3\ell}$ in the same way as in Lemma 3.1, but only for units $k \in [1\,..\lceil \frac{1}{2}\ell \rceil]$. This implies that only characters $S[k]$ of $S \in \mathcal{S}_{3\ell}$ with $k \leq \lceil \frac{1}{2}\ell \rceil$ are charged. It remains to show that any root-to-leaf path in $\mathcal{T}$ contains at most $3 + \log z$ light edges between a node at depth at least $\lfloor \frac{1}{2}\ell \rfloor$ and its child.

Consider a light edge from a node ν to its parent u at depth at least $\lfloor \frac{1}{2}\ell \rfloor$. Let ν' be a sibling of ν satisfying $\mathrm{size}(\nu') \geq \mathrm{size}(\nu)$, and let S_ν, $S_{\nu'}$ be the labels of the paths from the root to ν and ν', respectively. These labels differ on the last position only so, by Fact 2.1, they cannot be both periodic. Let $\tilde{\nu} \in \{\nu, \nu'\}$ be such that $S_{\tilde{\nu}}$ is not periodic, and let $\tilde{m} = \mathrm{size}(\tilde{\nu})$.

Consider the set $\mathcal{S}$ of length-ℓ strings corresponding to the leaves in the subtree of $\mathcal{T}$ rooted at $\tilde{\nu}$ (i.e., the labels of the root-to-leaf paths passing through $\tilde{\nu}$). Define $\overline{\mathcal{S}} := \{\overline{P} : P \in \mathcal{S}\}$ and note that $\overline{\mathcal{S}} \subseteq \mathcal{S}_\ell$, because $\mathcal{T}$ is the trie of *reversed* strings from $\mathcal{S}_\ell$. Let $e_1 < \cdots < e_{\tilde{m}}$ denote the ending positions of the leftmost occurrences in $T^\infty[1\,..)$ of strings in $\overline{\mathcal{S}}$. By definition, we have an occurrence of $\overline{S_{\tilde{\nu}}}$ ending in T^∞ at every position e_i with $i \in [1\,..\,\tilde{m}]$. Now, $\mathrm{per}(S_{\tilde{\nu}}) > \frac{1}{2}|S_{\tilde{\nu}}| \geq \frac{1}{4}\ell$ implies that $e_{i+1} - e_i > \frac{1}{4}\ell$ for every $i \in [1\,..\,\tilde{m}-1]$ (otherwise, the two close occurrences of $\overline{S_{\tilde{\nu}}}$ would yield $\mathrm{per}(\overline{S_{\tilde{\nu}}}) = \mathrm{per}(S_{\tilde{\nu}}) \leq \frac{1}{4}\ell$). Consequently, at least $\frac{1}{4}|\mathcal{S}| = \frac{1}{4}\tilde{m}$ length-ℓ substrings of $T^\infty[1\,..)$ have disjoint leftmost occurrences. Because each leftmost occurrence crosses or begins at a phrase boundary of the LZ77 parsing of T, we conclude that $z \geq \frac{1}{4}\tilde{m}$, and therefore $\text{size}(v) \leq \text{size}(\tilde{v}) = \tilde{m} \leq 4z$.

The reasoning above shows that once a root-to-leaf path encounters a light edge connecting a node u at depth at least $\lfloor \frac{1}{2}\ell \rfloor$ to its child v, we have $\text{size}(v) \leq 4z$. The number of the remaining light edges on the path is at most $\log(\text{size}(v)) \leq 2 + \log z$ by the standard bound applied to the subtree of $\mathcal{T}$ rooted at v. □

THEOREM 3.4. *Every string T of length n satisfies $r = \mathcal{O}(z \log z \max(1, \log \frac{n}{z \log z}))$.*

PROOF. To obtain tighter bounds on the number of irreducible LCP values in $[\ell \mathinner{.\,.} 2\ell)$, we consider three cases:

$\ell \leq \log z$. We repeat the proof of Lemma 3.1, except that we observe that the number of light edges on each root-to-leaf path in $\mathcal{T}$ is bounded by ℓ. Thus, the number of irreducible LCP values in $[\ell \mathinner{.\,.} 2\ell)$ is $\mathcal{O}(z\,\ell)$.

$\log z < \ell \leq \frac{n}{z}$. We use the bound $\mathcal{O}(z \log z)$ of Lemma 3.3.

$\frac{n}{z} < \ell$. We repeat the proof of Lemma 3.3, except that we observe that $|S_{3\ell}| \leq n$. Thus, the number of irreducible LCP values in $[\ell \mathinner{.\,.} 2\ell)$ is $\mathcal{O}(\frac{n \log z}{\ell})$.

The above upper bounds, applied for every $\ell = 2^i$ with $i \in [0 \mathinner{.\,.} \lfloor \log n \rfloor]$, yield

$$
\begin{aligned}
r &\leq \sigma + \sum_{i=0}^{\lfloor \log n \rfloor} \left| \left\{ j \in [2 \mathinner{.\,.} n] : \begin{array}{c} \mathrm{BWT}[j-1] \neq \mathrm{BWT}[j] \\ \mathrm{LCP}[j] \in [2^i \mathinner{.\,.} 2^{i+1}) \end{array} \right\} \right| \\
&= \mathcal{O}\left(\sigma + \sum_{i=0}^{\lfloor \log \log z \rfloor} z 2^i + \sum_{i=\lfloor \log\log z \rfloor + 1}^{\lfloor \log \frac{n}{z} \rfloor} z \log z + \sum_{i=\lfloor \log \frac{n}{z} \rfloor + 1}^{\lfloor \log n \rfloor} \frac{n \log z}{2^i} \right) \\
&= \mathcal{O}(\sigma + z \log z + z \log z \max(1, \log \tfrac{n}{z \log z}) + z \log z) \\
&= \mathcal{O}(z \log z \max(1, \log \tfrac{n}{z \log z})).
\end{aligned}
$$
□

3.3. Tight bounds in terms of δ

Let $\delta = \max_{m=1}^{n} \frac{1}{m} |\mathcal{S}_m|$ denote the (cyclic) *substring complexity* of T.[13] Note that letting $\delta = \sup_{m=1}^{\infty} \frac{1}{m} |\mathcal{S}_m|$ is equivalent because $|\mathcal{S}_m| \leq n$ holds for $m \geq 1$, which implies $\frac{1}{m}|\mathcal{S}_m| \leq 1 \leq |\mathcal{S}_1|$ for $m \geq n$. We start by noting that $\delta \leq z$ because $|\mathcal{S}_m| \leq mz$ holds for every $m \geq 1$, as observed in the proof of Lemma 3.1. Furthermore, $|\mathcal{S}_m| \leq m\delta$ holds by definition of δ, so δ can replace z in the proof of Lemma 3.1.

To adapt the proof Lemma 3.3, we need to generalize the observation that at most z substrings from $\mathcal{S}_\ell$ may have disjoint leftmost occurrences in $T^\infty[1 \mathinner{.\,.})$. This observation is easy because the LZ77 parsing naturally yields a set of z positions (phrase boundaries) in T. The substring complexity δ does not provide such structure, but as the lemma here implies, we can replace z by 3δ in the aforementioned observation. The proof of Lemma 3.5 is a straightforward modification of the argument used in Kociumaka et al.[13] (Lemma 6). For completeness, we write down the full reasoning, with technical details tailored to our notation (e.g., $\mathcal{S}_\ell$ defined in terms of T^∞ rather than T).

LEMMA 3.5 (based on Kociumaka et al.,[13] LEMMA 6). *For any integer $\ell \geq 1$, the total number of positions in $T^\infty[1 \mathinner{.\,.})$ covered by the leftmost occurrences of strings from $\mathcal{S}_\ell$ is at most $3\delta\ell$.*

PROOF. Let C denote the set of positions in $T^\infty[1 \mathinner{.\,.})$ covered by the leftmost occurrences of strings from $\mathcal{S}_\ell$, and let $C' = C \setminus [1 \mathinner{.\,.} \ell)$. For any $i \in C'$ denote $S_i = T^\infty[i-\ell+1 \mathinner{.\,.} i+\ell]$, and let $\mathcal{S} = \{S_i : i \in C'\} \subseteq \mathcal{S}_{2\ell}$. We will show that $|\mathcal{S}| = |C'|$. Let $i \in C'$. First, observe that, due to $i \geq \ell$, the fragment S_i is entirely contained in $T^\infty[1 \mathinner{.\,.})$. Furthermore, by definition, S_i contains the leftmost occurrence of some $S \in \mathcal{S}_\ell$. Thus, this occurrence of S_i in $T^\infty[1 \mathinner{.\,.})$ must also be the leftmost one in $T^\infty[1 \mathinner{.\,.})$. Consequently, the substrings S_i for $i \in C'$ are distinct.

We have thus shown that $|C'| = |\mathcal{S}| \leq |\mathcal{S}_{2\ell}|$. Because $|\mathcal{S}_{2\ell}| \leq 2\delta\ell$ holds by definition of δ, we obtain $|C| < |C'| + \ell \leq |\mathcal{S}_{2\ell}| + \ell \leq (2\delta+1)\ell \leq 3\delta\ell$. □

LEMMA 3.6. *For every $\ell \in [1 \mathinner{.\,.} n]$, the number of irreducible LCP values in $[\ell \mathinner{.\,.} 2\ell)$ is $\mathcal{O}(\delta \log \delta)$.*

PROOF. Compared to the proof of Lemma 3.3, we use the bound $|\mathcal{S}_{3\ell}| \leq 3\ell\delta$ instead of $|\mathcal{S}_{3\ell}| \leq 3\ell z$. The only other modification required is that, for every light edge connecting a node u of $\mathcal{T}$ at depth at least $\lfloor \frac{1}{2}\ell \rfloor$ to its child v, we need to prove $\text{size}(v) = \mathcal{O}(\delta)$.

Let $\tilde{m} \geq \text{size}(v)$ be defined as in the proof of Lemma 3.3. Recall that there are at least $\frac{\tilde{m}}{4}$ strings in $\mathcal{S}_\ell$ with disjoint leftmost occurrences in $T^\infty[1 \mathinner{.\,.})$. By Lemma 3.5, there are at most 3δ such substrings. Thus, $\text{size}(v) \leq \tilde{m} \leq 12\delta$. □

By replacing the thresholds $\log z$ and $\frac{n}{z}$ with $\log \delta$ and $\frac{n}{\delta}$, respectively, in the proof of Theorem 3.4, we immediately obtain a bound in terms of δ.

THEOREM 3.7. *Every string T of length n satisfies $r = \mathcal{O}(\delta \log \delta \max(1, \log \frac{n}{\delta \log \delta}))$.*

The following construction, analyzed in the full version of this paper, provides tight examples with substring complexity δ covering the whole spectrum between $\mathcal{O}(1)$ and $\Omega(\frac{n}{\log n})$. For $\ell \geq 1$ and $x \in [0 \mathinner{.\,.} 2^\ell)$, we set $\text{bin}_\ell(x) \in \{0, 1\}^\ell$ to be the binary representation of x.

LEMMA 3.8. *For all integers $\ell \geq 2$ and $K \geq 1$, the length n, the substring complexity δ, and the number of runs r in the BWT of a string $T_{\ell,K} \in \{\$, 0, 1, 2\}^+$, defined with*

$$
T_{\ell,K} = \left(\bigodot_{k=0}^{K-1} \bigodot_{i=0}^{2^\ell - 1} \left(2^{2^k \ell} \cdot \text{bin}_\ell(i) \right) \right) \cdot \$,
$$

satisfy $n = \Theta(2^{K+\ell}\ell)$, $\delta = \Theta(2^\ell)$, and $r = \Omega(2^\ell \ell K)$.

If $\delta = \Omega(\frac{n}{\log n})$, on the other hand, then a trivial upper bound $r = \mathcal{O}(n)$ is tighter than that of Theorem 3.7. In this case, the following construction, also analyzed in the full version of this paper, provides tight examples.

LEMMA 3.9. *For all integers $\ell \geq 2$ and $\Delta \in \Omega(2^\ell) \cap \mathcal{O}(2^\ell \ell)$, the length n, the substring complexity δ, and the number of runs r in the BWT of a string $T'_{\ell,\Delta} \in \{\$_1, \ldots, \$_\Delta, 0, 1, 2\}^+$, defined with*

$$
T'_{\ell,\Delta} = \left(\bigodot_{i=0}^{2^\ell - 1} \left(2^\ell \cdot \text{bin}_\ell(i) \right) \right) \cdot \$_1 \$_2 \ldots \$_\Delta,
$$

satisfy $n = \Theta(2^\ell \ell)$, $\delta = \Theta(\Delta)$, and $r = \Omega(2^\ell \ell)$.

Overall, combining Lemmas 3.8 and 3.9, we obtain the following tight lower bound for δ ranging from $\mathcal{O}(1)$ to $\Omega(n)$.

THEOREM 3.10. *For every $N \geq 1$ and $\Delta \in [1..N]$, there exists a string T whose length n, substring complexity δ, and the number of runs r in the BWT satisfy $n = \Theta(N)$, $\delta = \Theta(\Delta)$, and $r = \Theta(\min(n, \delta \log \delta \max(1, \log \frac{n}{\delta \log \delta})))$.*

3.4. Further combinatorial bounds

By combining Theorem 3.7 with known properties of the substring complexity δ, we obtain the first bound relating the number of BWT runs in the string and its reverse. No such bounds (even polynomial in $r \log n$) were known before.

COROLLARY 3.11. *If r and $\overline{r}$ denote the number of runs in the BWT of a length-n text and its reverse, respectively, then $\overline{r} = \mathcal{O}(r \log r \max(1, \log \frac{n}{r \log r}))$.*

PROOF. Because the value of δ is the same for the text and its reverse, Theorem 3.7 yields $\overline{r} = \mathcal{O}(\delta \log \delta \max(1, \log \frac{n}{\delta \log \delta}))$. Combining the results of Kempa and Prezza[12] (Theorem 3.9) and Kociumaka et al.[13] (Lemma 2) gives $\delta \leq r$. Consequently, we obtain $\overline{r} = \mathcal{O}(r \log r \max(1, \log \frac{n}{r \log r}))$. □

In the full version of this paper, we also characterize the sum of all irreducible LCP values:

THEOREM 3.12. *For every string of length n, the sum r_Σ of all irreducible LCP values satisfies $r_\Sigma = \mathcal{O}(n \log \delta)$.*

Due to $\delta \leq r$, the presented upper bound is always (asymptotically) at least as strong as the previous bound $r_\Sigma \leq n \log r$ by Kärkkäinen et al.[9] Furthermore, it can be strictly stronger because $\log \delta = o(\log r)$ is possible when $\delta = \log^{o(1)} n$. In the full version of this paper, we show that the strings of Lemmas 3.8 and 3.9 yield a matching lower bound:

THEOREM 3.13. *For every $N \geq 1$ and $\Delta \in [1..N]$, there exists a string T whose length n, substring complexity δ, and sum r_Σ of irreducible LCP values satisfy $n = \Theta(N)$, $\delta = \Theta(\Delta)$, and $r_\Sigma = \Theta(n \log \delta)$.*

4. FROM LZ77 TO RUN-LENGTH BWT

In this section, we outline our algorithm that, given the LZ77 parsing of a text $T \in \Sigma^n$, computes its run-length compressed BWT in $\mathcal{O}(z \text{ polylog } n)$ time. We start with an overview that explains the key concepts. Next, we present two new data structures utilized in our algorithm: the compressed string synchronizing set (Section 4.1) and the compressed wavelet tree (Section 4.2). The sketch of the final algorithm is then presented in Section 4.3.

For any substring Y of T^∞, we define

$$\text{lpos}(Y) = \min\{i \in [1..n] : T^\infty[i..i+|Y|) = Y\}.$$

We say that a substring Y of T^∞ is *left-maximal* if there exist distinct symbols $a, b \in \Sigma$ such that the strings aY and bY are also substrings of T^∞. The following definition, assuming $\Sigma \cap \mathbb{N} = \emptyset$, plays a key role in our construction.

DEFINITION 4.1 (BWT MODULO ℓ). *Let $T \in \Sigma^n$, $\ell \geq 1$ be an integer, and $Y_i = T^\infty[\text{SA}[i]..\text{SA}[i]+\ell)$ for $i \in [1..n]$. We define the string $\text{BWT}_\ell \in (\Sigma \cup \mathbb{N})^n$, called the* BWT modulo ℓ *(of T), as follows. For $i \in [1..n]$,*

$$\text{BWT}_\ell[i] = \begin{cases} \text{lpos}(Y_i) & \textit{if } Y_i \textit{ is left-maximal}, \\ \text{BWT}[i] & \textit{otherwise}. \end{cases}$$

The algorithm runs in $k = \lceil \log n \rceil$ rounds. For $q \in [0..k)$, the input to the qth round is $\text{RL}(\text{BWT}_{2^q})$ and the output is $\text{RL}(\text{BWT}_{2^{q+1}})$. At the end of the algorithm, we have $\text{RL}(\text{BWT}_{2^k}) = \text{RL}(\text{BWT})$, because $X \in \mathcal{S}_{2^k}$ is never left-maximal for $2^k \geq n$.

Informally, in round q, we are given a (run-length compressed) subsequence of BWT that can be determined based on sorting the suffixes only up to their prefixes of length 2^q. Notice that $\text{BWT}_\ell[b..e] \in \Sigma^*$ implies $\text{BWT}_{\ell+1}[b..e] \in \Sigma^*$ (because a prefix of a left-maximal substring is left-maximal). Hence, these subsequences need not be modified until the end of the algorithm (except possibly merging their runs with adjacent runs). For the remaining positions, BWT_ℓ identifies the (leftmost occurrences of) substrings to be inspected in the qth round with the aim of replacing their corresponding runs in BWT_ℓ with previously unknown BWT symbols (as defined in $\text{BWT}_{2\ell}$).

We call a block $\text{BWT}[b..e]$ *uniform* if all symbols in $\text{BWT}[b..e]$ are equal and *nonuniform* otherwise. The following lemma ensures the feasibility of the above construction.

LEMMA 4.2. *For any $\ell \geq 1$, it holds $|\text{RL}(\text{BWT}_\ell)| < 2r$.*

PROOF. Denote $\text{RL}(\text{BWT}_\ell) = ((c_1, \lambda_1), \ldots, (c_h, \lambda_h))$, letting $\lambda_0 = 0$. By definition of BWT_ℓ, if $c_i \in \mathbb{N}$, then the block $\text{BWT}(\lambda_{i-1}..\lambda_i]$ is nonuniform. Thus, there are at most $r - 1$ runs of symbols from $\mathbb{N}$ in BWT_ℓ.

On the other hand, $c_i \in \Sigma$ and $c_j \in \Sigma$, with $i < j$, cannot both belong to the same run in BWT. If this was true, then either $c_{i+1} \in \Sigma$ (which implies $c_{i+1} = c_i$, contradicting the definition of $\text{RL}(\text{BWT}_\ell)$), or $c_{i+1} \in \mathbb{N}$, which is impossible because $\text{BWT}(\lambda_i..\lambda_{i+1}]$ would then be nonuniform. Thus, there are at most r runs of symbols from Σ in BWT_ℓ. □

4.1. Compressed string synchronizing sets

Our algorithm builds on the notion of *string synchronizing sets,* recently introduced by Kempa and Kociumaka.[11] Synchronizing sets are one of the most powerful techniques for sampling suffixes. In the uncompressed setting, they are the key in obtaining time-optimal solutions to multiple problems, such as a state-of-the-art BWT construction algorithm for texts over small alphabets.[11] In this section, we introduce a notion of *compressed string synchronizing sets*. Our construction is the first implementation of synchronizing sets in the compressed setting and thus of independent interest.

We start with the definition of basic synchronizing sets.

DEFINITION 4.3 (τ-SYNCHRONIZING SET[11]). *Let $T \in \Sigma^n$ be a string and let $\tau \in \left[1..\left\lfloor \frac{n}{2} \right\rfloor\right]$ be a parameter. A set $\mathsf{S} \subseteq [1..n-2\tau+1]$ is called a* τ-synchronizing set *of T if it satisfies the following* consistency *and* density *conditions:*

1. *If $T[i..i+2\tau) = T[j..j+2\tau)$, then $i \in \mathsf{S}$ holds if and only if $j \in \mathsf{S}$ (for $i, j \in [1..n-2\tau+1]$),*

2. $\mathsf{S} \cap [i..i+\tau) = \emptyset$ *if and only if* $\mathrm{per}(T[i..i+3\tau-2]) \le \frac{1}{3}\tau$ *(for* $i \in [1..n-3\tau+2]$).

In most applications, we want to minimize $|\mathsf{S}|$. However, the density condition imposes a lower bound $|\mathsf{S}| = \Omega(\frac{n}{\tau})$ for strings of length $n \ge 3\tau - 1$ that do not contain substrings of length $3\tau - 1$ that are periodic with period at most $\frac{1}{3}\tau$. Therefore, we cannot hope to achieve an upper bound improving in the worst case upon the following one.

THEOREM 4.4 *(Kempa and Kociumaka[11]). For any string T of length n and parameter* $\tau \in \left[1..\left\lfloor \frac{n}{2} \right\rfloor\right]$, *there exists a τ-synchronizing set* S *of size* $|\mathsf{S}| = \mathcal{O}(\frac{n}{\tau})$. *Moreover, such* S *can be (deterministically) constructed in* $\mathcal{O}(n)$ *time.*

Storing S for compressible strings presents the following challenge: Although $|\mathsf{S}| = o(\frac{n}{\tau})$ *is* sometimes possible, it *is not* implied by $z \ll n$. For example, as noted by Bille et al.,[3] Thue–Morse strings satisfy $z = \mathcal{O}(\log n)$ yet they contain no periodic substring X with $\mathrm{per}(X) < \frac{1}{2}|X|$, and thus their synchronizing sets satisfy $|\mathsf{S}| = \Omega(\frac{n}{\tau})$. This prevents us from keeping the plain representation of S when $\tau = o(\frac{n}{z})$.

We therefore exploit a different property of compressible strings: That all their substrings Y satisfy $\mathrm{lpos}(Y) \in \bigcup_{j=1}^{z} (e_j - |Y| .. e_j]$, where e_j is the last position of the jth phrase in the LZ77 parsing of T. By consistency of S, it suffices to store $\bigcup_{j=1}^{z} \mathsf{S} \cap (e_j - 2\tau .. e_j]$. To decide whether $i \in \mathsf{S}$, we then locate $i' = \mathrm{lpos}(T[i..i+2\tau)$ and check if $i' \in \bigcup_{j=1}^{z} \mathsf{S} \cap (e_j - 2\tau .. e_j]$. This motivates the following (more general) definition.

DEFINITION 4.5 (COMPRESSED τ-SYNCHRONIZING SET). *Let* S *be a τ-synchronizing set of string* $T[1..n]$ *for some* $\tau \in \left[1..\left\lfloor \frac{n}{2} \right\rfloor\right]$, *and, for every* $j \in [1..z]$, *let* e_j *denote the last position of the jth phrase in the LZ77 parsing of T. For* $k \in \mathbb{Z}_{\ge 2}$, *we define the* compressed representation *of* S *as*

$$\mathrm{comp}_k(\mathsf{S}) := \bigcup_{j=1}^{z} \mathsf{S} \cap \left(e_j - k\tau .. e_j + k\tau\right].$$

In the full version of this paper, we prove that every text T has a synchronizing set S with a small compressed representation, and we show how to efficiently compute such S from the LZ77 parsing of T.

THEOREM 4.6. *There exists a Las-Vegas randomized algorithm that, given the LZ77 parsing of a string* $T \in \Sigma^n$, *a parameter* $\tau \in \left[1..\left\lfloor \frac{n}{2} \right\rfloor\right]$, *and a constant* $k \in \mathbb{Z}_{\ge 2}$, *constructs in* $\mathcal{O}(z \log^5 n)$ *time a compressed representation* $\mathrm{comp}_k(\mathsf{S})$ *of a τ-synchronizing set* S *of T satisfying* $|\mathrm{comp}_k(\mathsf{S})| \le 72kz$.

4.2. Compressed wavelet trees

Along with string synchronizing sets, wavelet trees,[8] originally invented for text indexing, play a central role in our algorithm. Unlike virtually all prior applications of wavelet trees, ours uses a sequence of very long strings (up to $\Theta(n)$ symbols). This approach is feasible because all strings are substrings of the text, which is stored in the LZ77 representation. In this section, we describe this novel variant of wavelet trees, dubbed here *compressed wavelet trees*. In particular, we prove an upper bound on their size, describe an efficient construction from the LZ77-compressed text, and show how to augment them to support some fundamental queries.

Let Σ be an alphabet of size $\sigma \ge 1$. Consider a string $W[1..m]$ over the alphabet Σ^ℓ so that W is a sequence of $m \ge 0$ strings of length $\ell \ge 0$ over the alphabet Σ. The wavelet tree of W is the trie $\mathcal{T}$ of Σ with each node v_X associated to a string $B_X \in \Sigma^*$ defined here based on W. We let $V(\mathcal{T}) = \bigcup_{d=0}^{\ell} \{v_X : X \in \Sigma^d\}$ denote the node-set of $\mathcal{T}$.

With each node $v_X \in V(\mathcal{T})$, we associate an increasing sequence $I_X[1..h]$ of *primary indices* such that

$$\{I_X[i] : i \in [1..h]\} = \{j \in [1..m] : W[j][1..|X|] = X\}.$$

Based on I_X, we define $B_X \in \Sigma^*$ so that, for $i \in [1..h]$,

$$B_X[i] = W[I_X[i]][|X| + 1]$$

if $|X| < \ell$, and $B_X = \varepsilon$ if $|X| = \ell$. In other words, B_X is a string containing the symbol at position $|X| + 1$ for each string of W that is prefixed by X. Importantly, the symbols in B_X occur in the same order as these strings occur in W.

As typically done in the applications of wavelet trees, we only explicitly store the strings B_X. The values of primary indices I_X are retrieved using additional data structures, based on the following observation.

LEMMA 4.7 (Grossi et al.[8]). *Let* $X \in \Sigma^d$, *where* $d \in [0..\ell)$. *For every* $c \in \Sigma$ *and* $j \in [1..|I_{Xc}|]$, *we have* $I_{Xc}[j] = I_X[i]$, *where* $B_X[i]$ *is the jth occurrence of c in* B_X.

We define the *compressed wavelet tree* $\mathcal{T}_c$ of W as the wavelet tree of W in which all strings B_X have been run-length compressed and, except $\{v_\varepsilon\} \cup \{v_{W[i]}\}_{i=1}^{m}$, all nodes v_X satisfying $|\mathrm{RL}(B_X)| \le 1$ have been removed (the unary paths are collapsed into single edges with their labels concatenated). The shape and edge labels of the resulting tree are identical to the *compacted trie* of $\{W[1], \ldots, W[m]\}$.

We store the edge labels of $\mathcal{T}_c$ as pointers to substrings in W. We assume that each edge of $\mathcal{T}_c$ and each element of $\mathrm{RL}(B_X)$ can be encoded in $\mathcal{O}(1)$ space. Because $|\mathrm{RL}(B_Y)| \ge 1$ holds for every internal node $v_Y \in V(\mathcal{T}_c)$ and, unless $|V(\mathcal{T}_c)| = 1$, each leaf v_Z in $\mathcal{T}_c$ can be injectively mapped to an element of $\mathrm{RL}(B_{Z'})$ for the parent $v_{Z'}$ of v_Z, the tree $\mathcal{T}_c$ uses $\mathcal{O}(1 + \sum_{v_X \in V(\mathcal{T}_c)} |\mathrm{RL}(B_X)|)$ space.

THEOREM 4.8. *If* $\mathcal{T}_c$ *is the compressed wavelet tree of a sequence W of length-ℓ strings, then* $\sum_{v_X \in V(\mathcal{T}_c)} |\mathrm{RL}(B_X)| = \mathcal{O}(1 + |\mathrm{RL}(W)| \log |\mathrm{RL}(W)|)$.

PROOF. Let $m = |W|$, $k = |\mathrm{RL}(W)| \le m$, and $k' = |\{W[i] : i \in [1..m]\}|$. Due to $|V(\mathcal{T}_c)| \le 1 + 2k' = \mathcal{O}(1 + k)$, we can focus on nodes $v_X \in V(\mathcal{T}_c)$ such that $|\mathrm{RL}(B_X)| \ge 2$.

The proof resembles that of Lemma 3.1. With each $X \in \Sigma^*$ such that $|\mathrm{RL}(B_X)| \ge 2$, we associate $|\mathrm{RL}(B_X)| - 1$ units of cost and charge them to individual elements of W. We then show that each run in $\mathrm{RL}(W)$ is in total charged at most $2 \log k'$ units of cost. Consequently,

$$\sum_{\substack{v_X \in V(\mathcal{T}_c) \\ |\mathrm{RL}(B_X)| \ge 2}} |\mathrm{RL}(B_X)| \le 4k \log k' = \mathcal{O}(k \log k).$$

Consider $X \in \Sigma^d$ with $|\mathrm{RL}(B_X)| \geq 2$; note that $d < \ell$. Let $\mathrm{RL}(B_X) = ((c_1, \lambda_1), \ldots, (c_h, \lambda_h))$. Observe that if we let $p_0 = I_X[\lambda_i]$ and $p_1 = I_X[\lambda_i + 1]$ for some $i \in [1 \mathinner{.\,.} h)$, then $W[p_0][d + 1] = c_i \neq c_{i+1} = W[p_1][d + 1]$. Moreover, $B_X[\lambda_i] \neq B_X[\lambda_i + 1]$ implies $W[p_0 + 1] \neq W[p_0]$ and $W[p_1 - 1] \neq W[p_1]$. The ith unit of cost is charged to $W[p_t]$, where $t \in \{0, 1\}$ is chosen depending on the sizes of subtrees of $\mathcal{T}_c$ rooted at the children of v_X, so that the subtree containing $v_{W[p_t]}$ has at most as many leaves as the subtree containing $v_{W[p_{1-t}]}$.

Now, consider a run $W[b \mathinner{.\,.} b'] = Y^\delta$ in $\mathrm{RL}(W)$. For a single depth d, the run could be charged at most twice, with at most one unit assigned to $W[b]$ due to $p_1 = b$ and at most one unit assigned to $W[b']$ due to $p_0 = b'$, both for $X = Y[1 \mathinner{.\,.} d]$. Moreover, note that the subtree size on the path from v_Y to the root v_ε of $\mathcal{T}_c$ doubles for every depth d for which the run was charged. Thus, the total charge of the run is at most $2 \log k'$ units. □

The key operation that we want to support on $\mathcal{T}_c$ is to compute the value $I_X[q]$ given $q \in [1 \mathinner{.\,.} |I_X|]$ and a pointer to $v_X \in V(\mathcal{T}_c)$. Let $W[1 \mathinner{.\,.} m]$ be a sequence of substrings of T^∞ of common length ℓ. Notice that if we have access to T, then the sequence W can be encoded in $\mathcal{O}(1 + |\mathrm{RL}(W)|)$ space. Namely, it suffices to store the length ℓ and the sequence $\mathrm{RL}((\mathrm{lpos}(W[i]))_{i \in [1 \mathinner{.\,.} m]})$. In the full version of this paper, we show that, given such encoding of W and the LZ77 parsing of T, the compressed wavelet tree of W supporting fast primary index queries can be constructed efficiently.

Theorem 4.9. *Given the LZ77 representation of $T[1 \mathinner{.\,.} n]$ and a sequence $W[1 \mathinner{.\,.} m]$ of $m \leq n$ same-length substrings of T^∞, represented as $\mathrm{RL}((\mathrm{lpos}(W[i]))_{i \in [1 \mathinner{.\,.} m]})$, the compressed wavelet tree of W, supporting primary index queries in time $\mathcal{O}(\log^4 n)$, can be constructed in time $\mathcal{O}((z + |\mathrm{RL}(W)|) \log^2 n)$.*

4.3. The algorithm

We are now ready to sketch the procedure constructing the sequences $\mathrm{RL}(\mathrm{BWT}_{2^q})$, where $q \in [0 \mathinner{.\,.} \lceil \log n \rceil]$.

Let $q \in [4 \mathinner{.\,.} \lceil \log n \rceil]$ (values $q \in [0 \mathinner{.\,.} 3]$ are handled separately) and let $\ell = 2^q$. We show how to compute $\mathrm{RL}(\mathrm{BWT}_{2\ell})$ given $\mathrm{RL}(\mathrm{BWT}_\ell)$ and the LZ77 parsing of T. The main idea of the algorithm is as follows.

Let S be a τ-synchronizing set of T, where $\tau = \left\lfloor \frac{\ell}{3} \right\rfloor$. As noted earlier, $\mathrm{BWT}_\ell[j] \in \Sigma$ implies $\mathrm{BWT}_{2\ell}[j] \in \Sigma$. Let $\mathrm{BWT}_\ell[y \mathinner{.\,.} y'] \in \mathbb{N}^+$ be a run in BWT_ℓ. By definition of BWT_ℓ, the suffixes of T^∞ starting at positions $i \in \mathrm{SA}[y \mathinner{.\,.} y']$ share a common prefix of length $\ell \geq 3\tau$. Thus, assuming that $\mathsf{S} \cap [i \mathinner{.\,.} i + \tau) = \emptyset$ holds for all $i \in \mathrm{SA}[y \mathinner{.\,.} y']$ (the periodic case is handled separately), by the consistency of S, all text positions $i \in \mathrm{SA}[y \mathinner{.\,.} y']$ share a common offset Δ with $i + \Delta = \min(\mathsf{S} \cap [i \mathinner{.\,.} i + \tau))$. This lets us deduce the order of length-2ℓ prefixes $T[i \mathinner{.\,.} i+2\ell)$ based on the order of strings $T[i + \Delta \mathinner{.\,.} i + 2\ell)$ starting at synchronizing positions. For this, from the sorted list of fragments $T[s \mathinner{.\,.} s + 2\ell)$ across $s \in \mathsf{S}$, we extract, using a wavelet tree, those preceded by $T[i \mathinner{.\,.} i + \Delta)$ (a prefix common to $T^\infty[i \mathinner{.\,.})$ for $i \in \mathrm{SA}[y \mathinner{.\,.} y']$). Importantly, the synchronizing positions s sharing $T[s - \tau \mathinner{.\,.} s + 2\ell)$ can be processed together; hence, by Theorem 4.6, applied for $k = 7$ so that $2\ell \leq k\tau$, it suffices to use $\mathcal{O}(z)$ distinct substrings.

In the full version of the paper, we provide the details of the above construction and obtain the following result.

Theorem 4.10. *There exists a Las-Vegas randomized algorithm that, given the LZ77 parsing of a text T of length n, computes its run-length compressed Burrows-Wheeler transform in time $\mathcal{O}((r + z) \log^6 n) = \mathcal{O}(z \log^8 n)$.*

Acknowledgments

This work was performed when the first author was a postdoctoral scholar at the University of California, Berkeley (supported by NSF grants no. 1652303 and 1934846, and an Alfred P. Sloan Fellowship grant), whereas the second author was a postdoctoral scholar at Bar-Ilan University, Israel (supported by ISF grants no. 1278/16 and 1926/19, a BSF grant no. 2018364, and an ERC grant MPM under the EU's Horizon 2020 Research and Innovation Programme, agreement no. 683064).

References

1. Amir, A., Iliopoulos, C.S., Radoszewski, J. Two strings at Hamming distance 1 cannot be both quasiperiodic. *Inf. Process. Lett. 128*, (2017), 54–57.
2. Belazzougui, D., Cunial, F., Gagie, T., Prezza, N., Raffinot, M. Composite repetition-aware data structures. In *CPM* (2015), Springer, Cham, Switzerland, 26–39.
3. Bille, P., Gagie, T., Gørtz, I.L., Prezza, N. A separation between RLSLPs and LZ77. *J. Discrete Algorithms 50*, (2018), 36–39.
4. Burrows, M., Wheeler, D.J. A block-sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation. Palo Alto, CA, 1994.
5. Gagie, T., Navarro, G., Prezza, N. Fully-functional suffix trees and optimal text searching in BWT-runs bounded space. arXiv 1809.02792 (2018).
6. Gagie, T., Navarro, G., Prezza, N. On the approximation ratio of Lempel–Ziv parsing. In *LATIN* (2018), Springer, Cham, Switzerland, 490–503.
7. Gagie, T., Navarro, G., Prezza, N. Fully functional suffix trees and optimal text searching in BWT-runs bounded space. *J. ACM 67*, 1 (2020), 1–54.
8. Grossi, R., Gupta, A., Vitter, J.S. High-order entropy-compressed text indexes. In *SODA* (2003), ACM/SIAM, Philadelphia, PA, USA, 841–850.
9. Kärkkäinen, J., Kempa, D., Piątkowski, M. Tighter bounds for the sum of irreducible LCP values. *Theor. Comput. Sci. 656*, (2016), 265–278.
10. Kempa, D. Optimal construction of compressed indexes for highly repetitive texts. In *SODA* (2019), SIAM, Philadelphia, PA, USA, 1344–1357.
11. Kempa, D., Kociumaka, T. String synchronizing sets: Sublinear-time BWT construction and optimal LCE data structure. In *STOC* (2019), ACM, New York, NY, USA, 756–767.
12. Kempa, D., Prezza, N. At the roots of dictionary compression: String attractors. In *STOC* (2018), ACM, New York, NY, USA, 827–840.
13. Kociumaka, T., Navarro, G., Prezza, N. Towards a definitive measure of repetitiveness. In *LATIN* (2020), Springer, Cham, Switzerland, 207–219.
14. Kreft, S., Navarro, G. On compressing and indexing repetitive sequences. *Theor. Comput. Sci. 483* (2013), 115–133.
15. Mäkinen, V., Belazzougui, D., Cunial, F., Tomescu, A.I. *Genome-Scale Algorithm Design: Biological Sequence Analysis in the Era of High-Throughput Sequencing*. Cambridge University Press, Cambridge, UK, 2015.
16. Manzini, G. An analysis of the Burrows-Wheeler transform. *J. ACM 48*, 3 (2001), 407–430.
17. Navarro, G. Indexing highly repetitive string collections, part I: Repetitiveness measures. *ACM Comput. Surv. 54*, 2 (2021), 1–31.
18. Navarro, G. Indexing highly repetitive string collections, part II: Compressed indexes. *ACM Comput. Surv. 54*, 2 (2021), 1–32.
19. Ohlebusch, E. *Bioinformatics Algorithms: Sequence Analysis, Genome Rearrangements, and Phylogenetic Reconstruction*. Oldenbusch Verlag, Bremen, Germany, 2013.
20. Ohno, T., Sakai, K., Takabatake, Y., Tomohiro, I, Sakamoto, H. A faster implementation of online RLBWT and its application to LZ77 parsing. *J. Discrete Algorithms*, (2018), 52-53:18–28.
21. Pevsner, J. *Bioinformatics and Functional Genomics*, 3rd edn. Wiley-Blackwell, Chichester, UK, 2015.
22. Policriti, A., Prezza, N. From LZ77 to the run-length encoded Burrows-Wheeler transform, and back. In *CPM* (2017), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 17:1–17:10.
23. Policriti, A., Prezza, N. LZ77 computation based on the run-length encoded BWT. *Algorithmica 80*, 7 (2018), 1986–2011.
24. Sirén, J., Välimäki, N., Mäkinen, V., Navarro, G. Run-length compressed indexes are superior for highly repetitive sequence collections. In *SPIRE* (2008), Springer, Berlin, Heidelberg, Germany, 164–175.
25. Ziv, J., Lempel, A. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory 23*, 3 (1977), 337–343.

Dominik Kempa ([kempa]@cs.jhu.edu), Johns Hopkins University, Baltimore, MD, USA.

Tomasz Kociumaka ([kociumaka]@berkeley.edu), University of California, Berkeley, CA, USA.

DOI:10.1145/3531446

To view the accompanying paper, visit doi.acm.org/10.1145/3531447

Technical Perspective
Computation Where the (inter)Action Is

By Jeffrey P. Bigham

SOUNDWATCH IS A prototype system that detects audio events and displays descriptions of them to deaf and hard-of-hearing people via the screen of their smartwatch. Beyond the system itself, SoundWatch contributes a case study of the opportunities and challenges we might expect as computation continues to move closer to where the interaction happens.

Access technology has long been a window into the future, and so we can learn a lot from prototypes like SoundWatch. As one example, speech recognition is now mainstream, but the people who have relied on it the longest are those who found it difficult to type otherwise. Mainstream user interfaces focus on a small set of modalities, whereas accessibility necessarily explores interactions beyond common ability assumptions.

Access technologies sense rather than try to understand. As an example, think about the difference between SoundWatch telling the user to "go open the door" versus alerting them that it might have heard a "doorbell." Both may lead the user to check the door, but the latter message better protects user agency and better enables a human user to make up for system limitations. If I don't have a doorbell or am not at home, I am better positioned to reason about what other sound events might be plausible if SoundWatch displays that it heard a doorbell.

Early access technologies were bulky. They also didn't work well. They were slow, inaccurate, and severely limited in what they could do. They were sometimes adopted nevertheless when they provided value over alternative approaches. Early sound recognition systems detected only a handful of sounds, plugged into the wall, and were expensive special purpose devices. These days, basic sound recognition comes standard on smartphones.

Computation, notably machine learning, has generally been moving closer to where the interaction happens. Smartphones made interaction mobile, and prototypes of wearable computers have existed for decades; commodity smartwatches might seem like a small step in comparison, but they significantly change interaction possibilities. A smartwatch screen is always glanceable, whereas a phone is often hidden away in a pocket or a bag. Phones must be intentionally carried around and are fragile, whereas a smartwatch is attached to people, making it difficult to forget or drop. Devices too bulky or too strange or too ugly get left behind (a well-known design consideration in accessibility), whereas smartwatches put computation in a 200-year-old, widely accepted form factor.

Smartwatches do not yet have the computational power necessary to be a person's only device, and ML models with good performance are big and computationally expensive. Model compression and research into efficient ML are tackling these issues. Regardless, at any given time, the best performing model for most interesting real-world problems will require more computation than the lowest-power device people regularly use. SoundWatch recognizes 20 sounds, but what if we wanted to recognize 1,000 or 10,000 sounds, transcribe speech, or be more robust to noise? Those capabilities will be available first on more powerful devices.

Computer scientists are comfortable designing architectures that trade off different computational capabilities and latencies. SoundWatch illustrates that this must now necessarily include multiple computational devices on our bodies. SoundWatch explored a network with a smartwatch, a smartphone and a remote server; what will it take to be prepared for the near future when these trade-offs must be considered relative to available computation on many wearable devices, for example, headphones, rings, contact lenses, and shoes worn by the user? The closer the computation is to the user interaction, the less powerful it likely is, which is a new human-centered trade-off where the objectives are not as easily or universally defined.

Human-computer interaction (HCI) research has a vital role to play in designing these architectures because deciding where computation should happen is not only a technical question but also a human one. The SoundWatch study provides an example. By interacting with SoundWatch, potential users were able to provide more ecologically valid feedback about which sounds needed to be detected quickly (for example, those related to safety), and which could reasonably take longer (for example, environmental sounds). A challenge not explored by SoundWatch is how to design usable systems whose performance and capabilities change when the underlying architecture changes (for example, when I have my phone versus when I don't). More HCI research is needed!

SoundWatch invites us to peer into near-term futures that will shape not only accessibility but broadly how we interact with technology. People may have been so focused waiting on a sci-fi vision of augmented humans that we have missed that computation is now pervasively with us. Everything from our phone, watch, left and right headphone, and many other devices are now capable of computation —what interactions could usefully be localized there, what performance will people expect, and what innovations across computer science will be needed to support them? SoundWatch is directly about improving accessibility, but prototypes like this ultimately push us to drive forward every area of computing.

Jeffrey P. Bigham is an associate professor in the Human Computer Interaction Institute at Carnegie Mellon University, Pittsburgh, PA, USA.

DOI:10.1145/3531447

To view the accompanying Technical Perspective, visit doi.acm.org/10.1145/3531446

SoundWatch: Deep Learning for Sound Accessibility on Smartwatches

By Dhruv Jain, Hung Ngo, Pratyush Patel, Steven Goodman, Khoa Nguyen, Rachel Grossman-Kahn, Leah Findlater, and Jon Froehlich

Abstract

Smartwatches have the potential to provide glanceable, always-available sound feedback to people who are deaf or hard of hearing (DHH). We present SoundWatch, a smartwatch-based deep learning application to sense, classify, and provide feedback about sounds occurring in the environment. To design SoundWatch, we first examined four low-resource sound classification models across four device architectures: watch-only, watch+phone, watch+phone+cloud, and watch+cloud. We found that the best model, VGG-lite, performed similar to the state of the art for nonportable devices although requiring substantially less memory (∼1/3rd) and that the watch+phone architecture provided the best balance among CPU, memory, network usage, and latency. Based on these results, we built and conducted a lab evaluation of our smartwatch app with eight DHH participants. We found support for our sound classification app but also uncovered concerns with misclassifications, latency, and privacy.

1. INTRODUCTION

Smartwatches have the potential to provide glanceable and always-available sound feedback to people who are deaf or hard of hearing (DHH) across multiple contexts.[3,5,17] A recent survey with 201 DHH participants[3] showed that, compared to smartphones and head-mounted displays (HMDs), a smartwatch is the most preferred device for nonspeech sound awareness due to privacy, social acceptability, and integrated support for both visual and haptic feedback.

Most prior work in wearable sound awareness, however, has focused on smartphones,[1,20] HMDs,[6,9] or custom wearable devices[13] that provide limited information (e.g., loudness) through a single modality (e.g., vision). For smartwatches specifically, studies have examined formative design prototypes for sound feedback,[5,17] but these prototypes have not included automatic sound classification—our focus. Furthermore, although recent deep learning research (e.g., see Jain et al.[11]) has examined models to automatically classify sounds, these cloud- or laptop-based models have high memory and processing power requirements and are unsuitable for low-resource portable devices.

Building on the above research, we present two smartwatch-based studies and a custom smartwatch-based application, called SoundWatch (see Figure 1). To design SoundWatch, we first quantitatively examined four state-of-the-art low-resource deep learning models for sound classification: *MobileNet, Inception, ResNet-lite,* and a quantized version of model used in *HomeSound*,[11] which we call VGG-lite, across four device architectures: watch-only, watch+phone, watch+phone+cloud, and watch+cloud. These approaches were intentionally selected to examine trade-offs in computational and network requirements, power efficiency, data privacy, and latency. Although direct comparison to prior work is challenging, our experiments show that the best classification model (VGG-lite) performed similar to the state of the art for nonportable devices although requiring substantially less memory (∼1/3rd). We also observed a strict accuracy-latency trade-off: the most accurate model was the slowest. Finally, we found that the two phone-based architectures (watch+phone and watch+phone+cloud) outperformed the watch-centric designs (watch-only and watch+cloud) in terms of CPU, memory, battery usage, and end-to-end latency.

Based on these quantitative experiments, we built SoundWatch and conducted a qualitative lab evaluation with eight DHH participants. SoundWatch incorporates the best-performing classification model from our system experiments (VGG-lite) and, for the purposes of evaluation, can be switched between all four device architectures. During the 90-min study session, participants used our prototype in three locations on a university campus (a home-like lounge, an office, and outdoors) and took part in a semistructured interview about their experiences, their views on accuracy-latency trade-offs and privacy, and ideas and concerns for future wearable sound awareness technology. We found that all participants generally appreciated SoundWatch across all contexts, reaffirming past sound awareness work.[3,5] However, misclassifications were concerning, especially outdoors because of background noise. For accuracy-latency trade-offs, participants wanted minimum delay for urgent sounds (e.g., car honk and fire alarms)—to take any required action—but maximum accuracy for nonurgent sounds (e.g., speech and background noise) to not be unnecessarily disturbed. Finally, participants selected watch+phone as the most preferred architecture due to privacy concerns with the cloud, versatility (no Internet connection required), and speed (watch+phone was faster than watch-only).

In summary, our work contributes (1) a comparison of deep learning models for sound classification on mobile devices; (2) a new smartwatch-based sound identification

The original version of this paper was published in *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, 2020.

Figure 1. Different use cases of our *SoundWatch* sound classification app and one of the four architectures (watch+phone).

system, called SoundWatch, with support for four device architectures; and (3) qualitative insights from in situ evaluation with eight DHH users, such as reactions to our designs, architectures, and ideas for future implementations.

This paper is based on our earlier ASSETS paper.[12] Since that paper was accepted in June 2020, much has changed. We released the SoundWatch codebase as open source (https://bit.ly/3bvgCLI) and our work helped guide subsequent literature (e.g., see Guo et al.[7]). The SoundWatch app is now available publicly on the Google Play Store (https://bit.ly/3bpEPTF, 500+ downloads to date). Additionally, sound recognition is integrated into both the major mobile platforms: Apple iOS and Google Android, demonstrating the impact of our work.

2. RELATED WORK

We situate our work within sound awareness needs, sound awareness tools, and sound classification research.

2.1. Sound awareness needs

Formative studies have examined the sounds, sound characteristics, and feedback modalities desired by DHH users. For sounds of interest, two large-scale surveys[1,3] showed DHH people most prefer urgent and safety-related sounds (e.g., sirens) followed by appliance alerts (e.g., microwave beep) and sounds about the presence of people (e.g., door knock calls). These preferences may be modulated by cultural factors: people who prefer oral communication may be more interested in some sounds (e.g., phone ring and conversations) than those who prefer sign language.[1,3]

In addition to these sounds, DHH users tend to desire information about certain sound characteristics (e.g., identity, location, and time of occurrence) more than others (e.g., loudness, duration, and pitch).[5,15] However, the utility of these characteristics may vary by location. For example, at home, awareness of a sound's identity and location may be sufficient,[10,11] but directional indicators are more important when mobile.[5] Besides location, social context (e.g., friends vs. strangers) could influence the use of the sound awareness tool,[3] and thus offering options for customization is key (e.g., using a sound-filtering menu).

In terms of feedback modalities, studies suggest combining visual and vibrational information for sound awareness[5, 17]; a smartwatch can provide both. Within the two modalities, prior work recommends using vibration to notify about sound occurrence and vision to show more information[1, 10]—which we also explore—although a recent study showed value in using complex vibration patterns to convey richer feedback (e.g., direction).[5]

We build on the above studies by examining the use of working smartwatch prototypes across contexts and revealing qualitative reactions and suggestions for system design.

2.2. Sound awareness technologies

Early research in sound awareness studied wrist-worn vibrotactile solutions, primarily to aid speech therapy by conveying voice tone[22] or frequency[21]; this work is complementary to our focus on nonspeech sound awareness. More recent work has examined stationary solutions for nonspeech sound awareness, such as on desktop displays.[15] Though useful for specific applications, these solutions are not conducive to multiple contexts. Toward portable solutions, Bragg et al.[1] and Sicong et al.[20] used smartphones to recognize and display sound identity (e.g., phone ringing and sirens). However, they evaluated their app in a single context (office[1] or a deaf school[20]) and focused on user interface rather than system performance—both are critical to user experience.

Besides smartphones, wearable solutions such as HMDs[6,9] and wrist-worn devices[13] have been examined. For example, Gorman[6] and Kaneko et al.[13] displayed the location of sound sources on an HMD and a custom wrist-worn device, respectively. We explore smartwatches to provide sound identity, the most desired sound property by DHH users.[1, 15] Although not specifically focused on smartwatches, Jain et al.[11] examined smartwatches as complementary alerting devices to smarthome displays that sensed and processed sound information locally and broadcasted it to the watches; we examine a self-contained smartwatch solution.

In summary, although prior work has explored sound awareness tools for DHH people, such as on portable devices,[6,9,13] this work has not yet built and evaluated a working smartwatch-based solution—a gap we address in our work.

2.3. Sound classification research

Early efforts in classifying sounds relied on handcrafted features such as zero-crossing rate, frame power, and pitch.[14,18] Though they performed reasonably well on clean sound files, these features fail to account for acoustic variations in the field (e.g., background noise).[14] More recently, machine learning-based classification has shown promise for specific field tasks such as gunshot detection or intruder alert systems.[4] For broad use cases, deep learning-based solutions have been investigated.[11,20] For example, Sicong et al.[20] explored a lightweight convolutional neural network (CNN) on smartphones to classify nine sounds preferred by DHH users (e.g., fire alarm and doorbell) in a school setting. Jain et al.[11] used deep CNNs running on a tablet to classify sounds in the homes of DHH users, achieving an overall accuracy of 85.9%. We closely follow the latter approach in our work by adapting it to low-resource devices (phone and watch) and performing evaluations in multiple contexts (home, work, and outdoors).

3. THE SOUNDWATCH SYSTEM

SoundWatch is an Android-based app designed for commercially available smartwatches to provide glanceable, always-available, and private sound feedback in multiple contexts. Building on previous work,[5,11] SoundWatch informs users about three key sound properties: identity, loudness, and time of occurrence through customizable visual and vibrational sound alerts (see Figures 1 and 3). We use a deep learning-based sound classification engine (running on the watch, paired phone, or cloud) to continually sense and process sound events in real time. Here, we describe our sound classification engine, our privacy-preserving sound sensing pipeline, system architectures, and implementation. Our codebase is open sourced: https://bit.ly/3bvgCLI.

3.1. Sound classification engine

To create a robust, real-time sound classification engine, we followed an approach similar to *HomeSound*,[11] which uses transfer learning to adapt a deep CNN–based image classification model (VGG) for sound classification. We downloaded three recently released (in Jan 2020) image-classification networks for small devices: *MobileNet,* 3.4MB; *Inception,* 41MB; and *ResNet-lite,* 178.3MB, and we used the quantized version of the network in HomeSound,[11] which we call VGG-lite, 281.8MB. We hypothesized that each network would offer different accuracy and latency trade-offs.

To perform transfer learning, similar to Jain et al.,[11] we used a large corpus of sound effect libraries—each of which provides a collection of high-quality, prelabeled sounds. Samples for 20 common sounds preferred by DHH people (e.g., dog bark, door knock, and speech)[1,3] were downloaded from six libraries—BBC, Freesound, Network Sound, UPC, TUT, and TAU. All sound clips were converted to a single format (16KHz, 16-bit, mono) and silences greater than one second were removed, resulting in 35.6 h of recordings. We divided the sound classes into three categories (see Table 1): high priority (containing the three most desired sounds by DHH people[1,15]); medium-priority sounds (10 sounds); and all sounds (20 sounds). Finally, we used the method by Hershey et al.[8] to compute log mel-spectrogram features for each category, which were then fed to the four networks, generating three models for each architecture (12 in total).

3.2. Sound sensing pipeline

For always-listening apps, privacy is a key concern. Although SoundWatch relies on a live microphone, we designed our sensing pipeline to protect user privacy. The system processes the sound locally on the watch or phone and, in the case of the cloud-based architectures, only uploads low-dimensional mel-spectrogram features. Although these features can be used to identify speech activity, the spoken content is challenging to recover. For signal processing, we take a sliding window approach: the watch samples the microphone at 16KHz and segments data into 1-second

Figure 2. A diagram of the four SoundWatch architectures with their sensing pipelines. Block widths are for illustration only and do not indicate actual computation time.

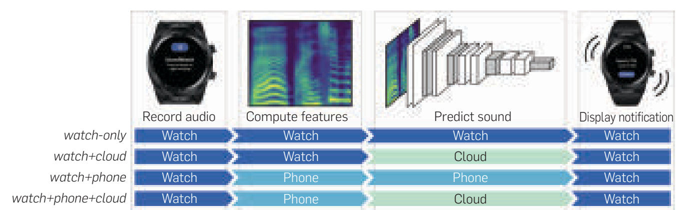


Figure 3. The SoundWatch user interface showing the (a) opening screen with a button to begin recording audio, (b) the notification screen with a "10-min" mute button, (c) the main app screen with more mute options, and (d) the paired phone app for customizing the list of enabled sounds.

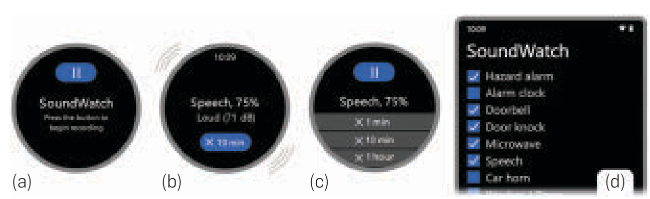


Table 1. The sounds and categories used to train our sound classification models.

All sounds ($N = 20$)	Fire/smoke alarm, alarm clock, door knock, doorbell, door-in-use, microwave, washer/dryer, phone ringing, speech, laughing, dog bark, cat meow, baby crying, vehicle running, car horn, siren, bird chirp, water running, hammering, drilling
High priority ($N = 3$)	Fire/smoke alarm, alarm clock, door knock
Medium priority ($N = 10$)	Fire/smoke alarm, alarm clock, door knock, doorbell, microwave, washer/dryer, phone ringing, car horn, siren, water running
Home context ($N = 11$)	Fire/smoke alarm, alarm clock, door knock, doorbell, door-in-use, microwave, washer/dryer, speech, dog bark, cat meow, baby crying
Office context ($N = 6$)	Fire/smoke alarm, door knock, door-in-use, phone ringing, speech, laughing
Outdoor context ($N = 9$)	Dog bark, cat meow, vehicle running, car horn, siren, bird chirp, water running, hammering, drilling

buffers (16,000 samples), which are fed to the sound classification engine. To extract loudness, we compute the average amplitude in the window. All sounds at or above 50% confidence and 45dB loudness are notified; others are ignored.

3.3. System architectures

We implemented four device architectures for SoundWatch: watch-only, watch+phone, watch+cloud, and watch+phone+cloud (see Figure 2). Because the sound classification engine (computing features and predicting sound) is resource intensive, the latter three architectures use a more powerful device (phone or cloud) for running the model. For only the cloud-based architectures, sound features are computed before being sent to the cloud to protect user privacy—that is, on the watch (watch+cloud) or on the phone (watch+phone+cloud). For communication, we use Bluetooth Low Energy (BLE) for watch-phone and WiFi or a cellular network for watch-cloud or phone-cloud.

3.4. User interface

For glanceability, we designed the SoundWatch app as a push notification; when a classified sound event occurs, the watch displays a notification along with a vibration alert. The display includes sound identity, classification confidence, loudness, and time of occurrence (see Figure 3). Importantly, each user can mute an alerted sound by clicking on the "10 min" mute button, or by clicking on the "open" button and selecting from a scroll list of mute options (1 min, 5 min, 10 min, 1 h, 1 day, or forever). Additionally, the user can filter alerts for any sounds using a customization menu on the paired phone app (see Figure 3d). Although future versions should run as an always-available service in Android, currently, the app must be explicitly opened on the watch (see Figure 3a). Once opened, the app runs continuously in the background.

4. SYSTEM EVALUATION

To assess the performance of our SoundWatch system, we performed two sets of evaluations: (1) a comparison of the four state-of-the-art sound classification models for small devices and (2) a comparison of the four architectures: watch-only, watch+phone, watch+cloud, and watch+phone+cloud. For all experiments, we used the Ticwatch Pro Android watch (4×1.2GHz, 1GB RAM) and the Honor 7x Android phone (8×2GHz, 3GB RAM). To emulate the cloud, we used an Intel i7 desktop running Windows 10 (4×2.5GHz, 16GB RAM).

4.1. Model comparison

We present our evaluation of classification accuracy and latency of the four models.

Accuracy. To calculate the "in-the-wild" accuracy of the models, we collected our own "naturalistic" dataset similar to *Home-Sound*.[11] We recorded 20 sound classes from nine locations (three homes, three offices, and three outdoors) using the same hardware as SoundWatch: Ticwatch Pro with a built-in microphone. For each sound class, we recorded three 10-second samples at three distances (5, 10, and 15 feet). When possible, we produced sounds naturally (e.g., by knocking or using a microwave). For certain difficult-to-produce sounds—such as a fire alarm—we played snippets of predefined videos on a laptop or phone with external speakers (total 54 videos were used). In total, we collected 540 recordings (~1.5 h).

Before testing our model, we divided our recordings into the three categories (all sounds, high priority, and medium priority) similar to our training set (see Table 1). For the medium- and high-priority test-sets, 20% of the sound data was from excluded categories that our models should ignore (called the "unknown" class). For example, the high-priority test-set included 20% recordings from outside of the three high-priority classes (fire/smoke alarm, alarm clock, and door knock).

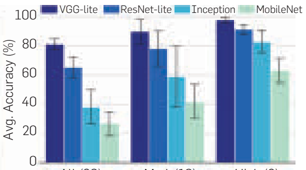
Figure 4 shows the results of classifying sounds in each category. Overall, VGG-lite performed best (avg. inference accuracy = 81.2%, *SD* = 5.8%) followed by ResNet-lite (65.1%, *SD* = 10.7%), Inception (38.3%, *SD* = 17.1%), and MobileNet (26.5%, *SD* = 12.3%); a one-way repeated measures ANOVA on all sounds yielded a significant effect of models on the accuracy ($F_{3,2156}$ = 683.9, $p < .001$). As expected, the inference accuracy increased as the number of sounds decreased from all (20 sounds) to medium (10 sounds) and high priority (3 sounds). In analyzing performance as a function of context, home and office outperformed outdoors for all models. With VGG-lite, for example, average accuracy was 88.6% (*SD* = 3.1%) for home, 86.4% (*SD* = 4.3%) for office, and 71.2% (*SD* = 8.2%) for outdoors. A post hoc inspection revealed that outdoor recordings incurred interference due to the background noise.

To assess interclass errors, we computed a confusion matrix for medium-priority sounds. Although per-class accuracy varied across models, microwave, door knock, and washer/dryer were consistently the best-performing classes, with VGG-lite achieving average accuracy of 100% (*SD* = 0), 100% (*SD* = 0), and 96.3% (*SD* = 2.3%), respectively. The worst-performing classes were more model dependent but generally included alarm clock, phone ring, and siren, with VGG-lite achieving 77.8% (*SD* = 8.2%), 81.5% (*SD* = 4.4%), and 88.9% (*SD* = 3.8%), respectively. For these poorly performing classes, understandable mix-ups occurred such as confusions among similar sounding events (e.g., alarm clocks and phone rings).

Latency. Low latency is crucial to achieving a real-time sound identification system. To evaluate model latency, we wrote a script to loop through the sound recordings in our dataset for 3 h (1080 sounds) and measured the time required to classify sounds from the input features on both the watch and the phone. Understandably, the latency increased with the model size: the smallest model, MobileNet, performed the fastest on both devices (*avg. latency* on watch: 256 ms, *SD* = 17 ms; phone: 52 ms, *SD* = 8 ms), followed by Inception (watch: 466 ms, *SD* = 15 ms; phone: 94 ms, *SD* = 4 ms), and

Figure 4. Average accuracy (and SD) of the four models for three sound categories and three contexts. Error bars in the graph show 95% confidence intervals.

	VGG-lite	ResNet-lite	Inception	MobileNet
All sounds	81.2 (5.8)	65.1 (10.7)	38.3 (17.1)	26.5 (12.3)
Med. Priority	89.6 (8.7)	78.1 (11.9)	59.0 (20.8)	41.8 (11.6)
High Priority	97.6 (1.7)	91.1 (3.4)	82.9 (7.6)	63.0 (8.5)
Home	88.6 (3.1)	71.3 (6.4)	41.7 (13.9)	30.5 (6.4)
Office	86.4 (4.3)	70.5 (6.4)	43.9 (11.7)	31.2 (8.9)
Outdoors	71.2 (8.2)	56.3 (15.3)	32.0 (23.2)	20.1 (14.7)



ResNet-lite (watch: 1615 ms, *SD* = 30 ms; phone: 292 ms, *SD* = 13 ms). VGG-lite, the largest model, was the slowest (watch: 3397 ms, *SD* = 42 ms; phone: 610 ms, *SD* = 15 ms).

Model comparison summary. In summary, for phone and watch models, we observed a strict accuracy–latency trade-off—for example, the most accurate model VGG-lite (*avg. accuracy* = 81.2%, *SD* = 5.8%) was also the slowest (*avg. latency* on watch: 3397 ms, *SD* = 42 ms). Further, the models MobileNet and Inception performed too poorly for practical use (*avg. accuracy* < 40%). ResNet-lite was in the middle (*avg. accuracy* = 65.1%, *SD* = 10.7%; *avg. latency* on watch: 1615 ms, *SD* = 30 ms).

Comparison to prior approach. We also evaluated the performance of the full VGG model running on the cloud, which is used in the state-of-the-art prior work on sound classification.[11] The average inference accuracy (84.4%, *SD* = 5.5%) was only slightly better than our best mobile-optimized model (VGG-lite, *avg.*= 81.2%, *SD* = 5.8%)—a promising result as our VGG-lite model is less than 1/3rd the size of VGG (281.8MB vs. 845.5MB).

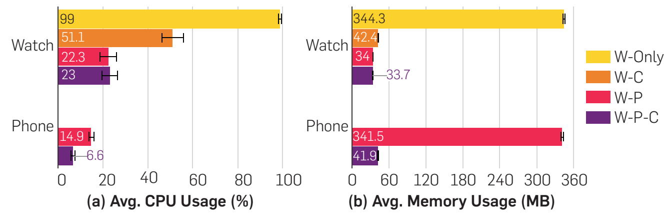
4.2. Architecture evaluation

We compared the performance of four different architectures of SoundWatch: watch-only, watch+phone, watch+cloud, and watch+phone+cloud (see Figure 2), which may differ in performance and usability.

For each architecture, we used the most accurate model on the watch and phone: VGG-lite; the cloud used the full VGG model. Informed by prior work,[16] we measured CPU, memory, network usage, end-to-end latency, and battery consumption. For the evaluation, we used a script running on a laptop that looped through the sound recordings for 3 h to generate sufficient sound samples (1080). For the battery experiment only, the script ran until the watch battery reached 30% or less (i.e., just above the 25% trigger for low-power mode), a common evaluation approach.[16] To determine CPU, memory, and network usage, we used *Android Profiler,* a commonly used profiling tool. For power usage, we used *Battery Historian*. Finally, to determine end-to-end latency, we measured the elapsed time (in milliseconds) between the start of the sound recording to when the notification is shown. Here, we detail our results.

CPU Utilization. Minimizing CPU use will maximize the smartwatch's battery performance and lower the impact on other running apps. Our results for CPU usage on the watch and phone are as shown in Figure 5a. As expected, the watch's CPU utilization was lowest for classifications performed on the phone (watch+phone; *avg.*= 22.3%, *SD* = 11.5%, *max* = 42.3%) or on the cloud (watch+phone+cloud; avg. = 23.0%, *SD* = 10.8%, *max* = 39.8%). In these architectures, the watch was used only for recording sounds and supporting user interactions. For watch+cloud, the watch additionally computed the sound features and communicated with the cloud over WiFi, which resulted in significantly higher CPU utilization (*avg.*= 51.1%, *SD* = 14.9%, *max* = 76.1%). Finally, for the watch-only design, CPU utilization nearly maxed out (*avg.*= 99.0%, *SD* = 2.1%, *max* = 100%) because the classification model ran directly on the watch, revealing that this design is impractical for real-world use. However, future advancements in machine learning and wearable technology may lead to smaller models and more powerful watches that can run these models locally.

Figure 5. Average CPU (a) and memory (b) usage of the four architectures. Error bars show 95% confidence intervals.



Memory usage. A smartwatch app must be memory efficient. Unsurprisingly, we found that the memory usage heavily depended on where the model (281.8MB) was running; hence, watch-only and watch+phone consumed the highest memory on the watch (*avg.* = 344.3MB, *SD* = 2.3MB, *max* = 346.1MB) and phone (*avg.*= 341.5MB, *SD* = 3.0MB, *max* = 344.1MB), respectively (see Figure 5b). This indicates that running a large model such as VGG-lite on the watch will likely exceed the memory capacity of current smartwatches. The other app processes (e.g., UI and computing features) required less than 50MB of memory.

Network usage. Low network usage increases the app portability, especially in low-signal areas, and may help reduce Internet costs. Only the cloud-based architectures required network because the classifications were performed locally for watch- or phone-based designs. Specifically, for watch+cloud, the average network consumption, when the system was actively classifying sounds every second, was 486.8B/s (*SD* = 0.5B/s, *max* = 487.6B/s), and for watch+phone+cloud, it was 486.5B/s (*SD* = 0.5B/s, *max* = 487.2B/s), which is very low (~1.8MB/h). In reality, sounds will likely not occur every second, which will reduce the total consumption even further.

Battery consumption. We measured the battery drain time from full charge until 30% (see Figure 6), finding that the watch-only architecture consumed a lot of battery: it reached 30% battery in 3.3 h only. Within the remaining architectures, both watch+phone (30% at 15.2 h) and watch+phone+cloud (30% at 16.1 h) were more efficient than watch+cloud (30% at 12.5 h), because the latter used WiFi, which consumes more energy than BLE.[19] Similar trends were observed on the phone; however, running the model on the phone (watch+phone) was still tolerable compared to the watch (see Figure 6). In summary, we expect the watch-only design would be impractical for daily use, whereas others are usable with the on-device implementations fairing slightly better than the cloud ones.

End-to-end latency. A real-time sound awareness system needs to be performant. Figure 7 shows a computational breakdown of end-to-end latency, that is, the total time taken in obtaining a notification for a produced sound. On average, watch+phone+cloud performed the fastest (*avg. latency* = 1.8 s, *SD* = 0.2 s) followed by watch+phone (*avg.*= 2.2 s, *SD* = 0.1 s), which needed more time for running the model on the phone (vs. cloud), and watch+cloud (*avg.*= 2.4 s, *SD* = 0.0 s), which required more time to compute features on the watch (vs. phone in watch+phone+cloud). As expected, watch-only

Figure 6. Battery level over time on the (a) watch and the (b) phone for the four architectures: watch only, watch+cloud, watch+phone, and watch+phone+cloud. Baseline represents the case without the SoundWatch app running.

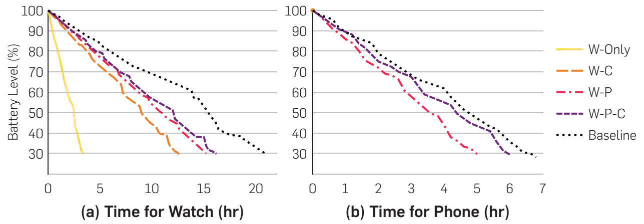
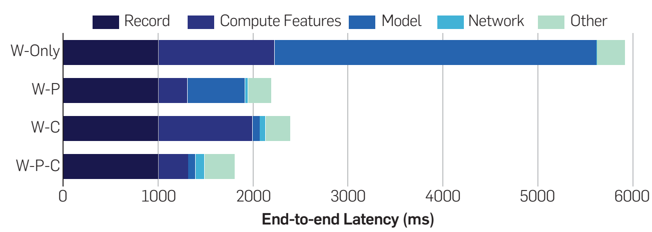


Figure 7. Breakdown of end-to-end latency for the four architectures.



was considerably slower (*avg*.= 5.9 s, *SD* = 0.1 s) and is, thus, currently unusable (though future smartwatches could be more capable). In summary, except for watch-only, all architectures had a latency of about 2 s; we evaluate whether this is acceptable in our user study.

Architecture evaluation summary. In summary, watch+phone and watch+phone+cloud outperformed the watch+cloud architecture for all system parameters. Additionally, the watch-only architecture was deemed impractical for real-life use due to high CPU, memory, and battery usage, and a large end-to-end latency. Among the phone-based architectures, watch+phone+cloud performed better than the watch+phone.

5. USER STUDY

To study end-user perceptions of our system results and reactions to SoundWatch across multiple contexts, we performed a lab and campus walkthrough evaluation with eight DHH participants. Although SoundWatch can support any architecture or model, we used only the best-performing architecture (watch+phone) and model (VGG-lite) for this study.

5.1. Participants

We recruited eight DHH participants (three women, three men, and two nonbinary) using email, social media, and snowball sampling. Participants were on average 34.8 years old (*SD* = 16.8, *range* = 20–63). Four had profound hearing loss, three had severe, and one had moderate. Seven reported onset as congenital and one reported one year of age. Seven participants used hearing devices: three used cochlear implants, one used hearing aids, and three used both. For communication, five participants preferred sign language and three preferred to speak verbally. All participants reported fluency with reading English (5/5 on rating scale, 5 is best). Participants received $40 as compensation.

5.2. Procedure

The in-person procedure took place on a university campus and lasted up to 90 min. Sessions were led by the lead author who is hard of hearing and knows level-2 ASL. A real-time transcriptionist attended all sessions and five participants opted to additionally have a sign language interpreter present. Questions were presented visually on an iPad, whereas responses and follow-up discussion were spoken or translated to/from ASL. The session began with a demographic questionnaire, followed by a three-part protocol:

Part 1: Introducing SoundWatch (5–10 min). First, we asked about general thoughts on using smartwatches for sound awareness. The researcher then demonstrated SoundWatch by explaining the UI and asking participants to wear the watch while making three example sounds (speech, door knock, and phone ring). Participants could also make their own sounds (e.g., by knocking or speaking).

Part 2: Campus walk (20–25 min). Next, the researcher and the participant (with the watch and phone) visited three locations on campus in a randomized order: (1) a home-like location (a building lounge), (2) an office-like location (a grad student office), and (3) an outdoor location (a bus stop). These locations allowed participants to experience SoundWatch in different auditory contexts. In each location, participants used the watch naturally (e.g., by sitting on a chair in an office) for about 5 min. In locations with insufficient sound activity (e.g., if the lounge was empty), the researcher produced some sounds (e.g., by washing hands or opening a door). Before exiting each location, participants completed a short feedback form.

Part 3: Posttrial interview (45–50 min). After the campus walk, participants returned to the lab for a semistructured interview about their overall experience, perceptions of SoundWatch across the three locations, reactions to the UI, and any privacy concerns. We then asked about specific technical considerations, such as accuracy–latency trade-offs and the four possible SoundWatch architectures. For accuracy–latency, we gathered their expectations for minimum accuracy and maximum delay and whether these perspectives changed based on sound type (e.g., urgent vs. nonurgent sounds) or context (e.g., home vs. office). To help discuss the four SoundWatch architectures—and to more easily allow our participants to understand and track differences—we prepared a chart enumerating key characteristics such as battery or network usage with a HIGH, MEDIUM, or LOW rating based on our system experiment findings. Finally, we asked participants to rate the "ease-of-use" of each architecture (high, med, or low) by weighing factors such as the Internet requirement, number of devices to carry (e.g., 1 for watch-only vs. 2 for watch+phone), and the size of visual display (e.g., small for watch vs. medium for phone) and provide reasoning for their choice.

5.3. Data analysis

We analyzed the interview transcripts and the in situ form responses using an iterative coding approach.[2] To begin, we randomly selected three out of eight transcripts; two researchers read these transcripts and developed an initial codebook. The researchers then independently assigned codes to the three transcripts, while simultaneously refining their own copy of the codebook (adding, merging, or deleting codes). The researchers then met again to discuss and refine the codebook, resulting in 12 level-1 codes and 41 level-2 codes) arranged in a hierarchy. This final codebook was used to code the remaining five transcripts by the two coders, resulting in an interrater agreement (measured using Krippendorff's alpha) of 0.79 (*SD* = 0.14, *range* = 0.62-1) and a raw agreement of 93.8% (*SD* = 6.1%, *range* = 84.4%-100). Conflicting code assignments were resolved via consensus.

5.4. Findings

We detail participants' experience with SoundWatch during the campus walk as well as comments on model accuracy–latency, system architectures, and the user interface.

Experience with campus walk. All participants found the watch generally useful to help with the everyday activities in all three contexts (home-like lounge, office, and outdoors). For example,

> *"My wife and I tend to leave the water running all the time so this app could be beneficial and save on water bills. It was helpful to know when the microwave beeps instead of having to stare at the time [microwave display]." (P6)*
> *"This is very useful for desk type work situations. I can use the watch to help alert me if someone is knocking the door, or coming into the room from behind me." (P7)*

However, all participants also reported problems, the most notable being delay and misclassifications; the latter were higher in outdoor contexts than in others. For example,

> *"The app is perfect for quiet settings such as [at] home. [While outdoors,] some sounds were misinterpreted, such as cars were recognized as water running." (P3)*

In situ feedback form responses corroborate these comments, with average usefulness for lounge (4.8/5 on a rating scale (5 is best), *SD* = 0.4) and office (4.6/5, *SD* = 0.5) being higher than for outdoors (3.5/5, *SD* = 0.5).

Even with a low usefulness rating in outdoor settings, all participants wanted to use the app outdoors, mentioning that they may be able to use contextual information to supplement inaccurate feedback. For example,

> *"Sure there were some errors outdoors, but it tells me sounds are happening that I might need to be aware of, so I can look around and check my environment for cues." (P8)*

Model accuracy–latency comparison. Deep learning-based sound recognition will never be 100% accurate. Thus, we asked participants about the minimum required accuracy and the maximum tolerable delay at which they will use a smartwatch app. The most common preference was a maximum delay of *"five seconds"* (5/8) and a minimum accuracy of 80% (6/8); however, this choice was additionally modulated by the specific sound type. Specifically, for urgent sounds (e.g., fire alarms or car horn), participants wanted the minimum possible delay (at the cost of accuracy) to get quick information for any required action, because *"I'll at least know something is happening around me and [...] can look around to see if a car is honking"* (P2).

In contrast, for nonurgent sounds (e.g., speech and laughing), more accuracy was preferred because participants mentioned that repeated errors could be annoying (7/8). For example:

> *"I don't care about speech much, so if there is a conversation, well fine, doesn't matter if I know about it 1-2 second later or 5 seconds later, does it? But if it makes mistakes and I have to get up and check who is speaking every time it makes a mistake, that can be really frustrating." (P5)*

Finally, for medium-priority sounds (e.g., microwave for P3), participants (7/8) wanted a balance, tolerating a moderate amount of delay for moderate accuracy.

Besides sound type, preference also varied with the context of use (home vs. office vs. outdoors). Participants preferred having less delay in more urgent contexts and vice versa. *That is,* for the home, participants (8/8) wanted high accuracy—and accepted more delay—because, for example:

> *"I know most of what is going on around my home. And at home, I am generally more relaxed [so] delay is okay. But, I don't want to be annoyed by errors in my off time." (P8)*

For the office, participants (6/8) felt they would tolerate a moderate level of accuracy with a moderate level of delay, because *"something may be needing my attention but it's likely not a safety concern to be quick about it"* (P8). Preferences for outdoors were split: four participants wanted a minimum delay (at the cost of accuracy), but the other four did not settle for a single response, mentioning that the trade-off would depend on the urgency of the specific sound:

> *"If it's just a vehicle running on the road while I am walking on the sidewalk, then I would want it to only tell if it's sure that it's a vehicle running, but if a car is honking say if it behind me, I would want to know immediately." (P2)*

Architecture comparison. By saliently introducing the performance metrics (e.g., battery usage) and usage requirements (e.g., Internet connection for cloud), we gathered qualitative preferences for the four possible SoundWatch architectures: watch-only, watch+phone, watch+cloud, and watch+phone+cloud.

In general, watch+phone was the most preferred architecture among all participants, because, compared to watch-only, it is faster, requires less battery, and has more visual state available for customization. In addition, compared to cloud-based designs, watch+phone is more private and self-contained (does not need Internet).

However, five participants wanted the option to be able to customize the architecture on the go, mentioning that in outdoor settings, they would instead prefer to use watch+phone+cloud because of speed and accuracy advantages. This is because in the outdoor context, data privacy is of less concern for them. For example:

> *"Accuracy problems could be more [outdoors] due to background noise and [thus] I prefer to use cloud for [stronger] models if [the] internet is available. At home/office, there is a possibility of private data breach." (P6)*

Watch+cloud was preferred by two participants only for cases where it is hard to carry a phone, such as in a *"gym or [while] running outdoors"* (P1). Finally, watch-only was not preferred for any situation because of high battery drain.

User interface suggestions. Overall, participants appreciated the minimalistic app design and the customization options (mute button and checklist on phone). When asked about future improvements, they suggested three: (1) show the urgency of sounds—for example, using vibration patterns or visual colors; (2) show direction of sounds, particularly for outdoor contexts; and (3) explore showing multiple sounds to compensate for inaccuracy:

> *"You could give suggestions for what else sound could be when it's not able to recognize. For example, [...] if it is not able to tell between a microwave and a dishwasher, it could say "microwave or dishwasher", or at least give me an indication of how it sounds like, you know like a fan or something, so I can see and tell, oh yeah, the dishwasher is running." (P4)*

6. DISCUSSION

Our work reaffirms DHH users' needs and preferences for smartwatch-based sound awareness[5, 17] but also (1) implements and empirically compares state-of-the-art deep learning approaches for sound classification on smartwatches, (2) contributes a new smartwatch-based sound identification system with support for multiple device architectures, and (3) highlights DHH users' reactions to accuracy–latency trade-offs, device architectures, and potential concerns. Here, we reflect on further implications and limitations of our work.

6.1. Utility of sound recognition

How well does sound recognition tool need to perform to provide value? Our findings show that this is a complex question that requires further study. Although improving overall accuracy, reducing latency, and supporting a broad range of sound classes is clearly important, participants felt that urgent sounds should be prioritized. Thus, we wonder, would an initial sound awareness app that supports three to ten urgent sounds be useful? One way to explore this question is to study SoundWatch—or a similar app—over a longitudinal period with multiple customization options. However, this approach also introduces ethical and safety concerns as automatic sound classification will never be 100% accurate. High accuracy on a limited set of sounds could (incorrectly) gain the user's trust, and the app's failure to recognize a safety sound (e.g., a fire alarm) even once could be dangerous. In general, a key finding of our research is that users desire customization (e.g., which sounds to classify) and transparency (e.g., classification confidence).

6.2. Toward improving accuracy

Our user study suggests a need to further improve system accuracy or at least explore other ways to mitigate misclassification. One possibility, suggested by P4, is to explore showing multiple "possible" sounds instead of the most probable sound—just as text autocomplete shows n-best words. Another idea is to sequentially cascade two models, using the faster model to classify a small set of urgent sounds and the slower model for lower-confidence classifications and less-urgent sounds. End-user customization should also be examined. Each user could select the desired sounds and the required accuracy, and the app could dynamically fine-tune the model (e.g., by using a weighted class accuracy metric). Finally, as proposed by Bragg et al.,[1] researchers should explore end user interactive training of the model. Here, guided by the app, participants could record sounds of interest to either improve existing sound classes or to add new ones. Of course, this training may be tedious and difficult if the sound itself is inaccessible to a DHH user.

6.3. Privacy implications

Our participants were concerned with how cloud-based classification architectures may invade their own "sound" privacy and of others around them. However, uploading and storing data on the cloud have benefits. These datasets can be used for improving the classification model. Indeed, modern sound architectures on IoT devices (e.g., Alexa and Siri) use the cloud for exchanging valuable data. A key difference to our approach is that these devices only transmit after listening to a trigger word. Thus, what are the implications for future always-listening sound awareness devices? We see three. First, the users should have control over what data gets uploaded, which can be customized based on context (e.g., offices might have more private conversations than outdoors). Second, future apps will need clear privacy policies such as GDPR or CCPA that outline how and where the data is stored and what guarantees the users have. Finally, users should always have access to their data with an option to potentially delete it, in entirety, from the cloud.

6.4. Future smartwatch applications

In contrast to past wearable sound awareness work,[6,9,13] we used commercially available smartwatches, a mainstream popular device that is more socially acceptable than HMDs[6,9] or custom hardware-based[6] solutions—and that may be preferred over smartphones for sound recognition feedback.[3] So, what are other compelling applications of a smartwatch-based sound awareness for DHH users? Full speech transcription, a highly desired feature by DHH users,[3] is difficult to accommodate on the small watch screen, but future work could explore highlighting important keywords or summarizing key conversation topics. Sound localization is also desired[1,5] and could be investigated by coupling the watch with a small external microphone array or designing a custom watch with multiple microphones. However, how best to combine different sound and speech features (e.g., topic summary, direction, and identity) on the watch is an open question. Goodman et al.[5] investigated designs for combining sound identity, direction, and loudness on watch; however, this study was formative with a focus on user interface. Future work should also explore the system design aspects of showing multiple features—a challenging problem given the smartwatch's low-resource constraints.

6.5. Limitations

First, although our sound recognition technology is heavily informed by DHH perspectives, such as those of our hard-of-hearing lead author, we do not assume it is universally desired. Some DHH people may feel negatively toward this technology, especially those who identify as part of deaf culture.[1,3] At the same time, past work,[1,3] such as our own survey with 201 DHH participants,[3] suggests the DHH community is broad and many DHH individuals do find sound recognition valuable. Still, future work should continue to examine preferences for sound feedback with a diverse section of the DHH population to verify our findings.

Second, our short 20-min campus walk, although useful as an initial, exploratory study, could not investigate pragmatic issues, such as user perception of battery life and long-term usage patterns. Future work should perform a longitudinal deployment and compare results with our lab findings.

Third, our model accuracy results, though gathered on real-life recordings of 20 sounds, do not accurately reflect real-world use where other sounds beyond those 20 could also occur. Although our approach provides a baseline for model comparison and contextualizing user study findings, a more accurate experiment would include a post hoc analysis of sound data collected from longitudinal watch use.

Finally, we evaluated our models on specific hardware devices (Ticwatch Pro Watch, Honor 7x Phone). Although the relative comparisons are likely generalizable, the absolute performance metrics will change as the mobile and wearable technologies evolve in the future. Additional studies will be needed then.

7. CONCLUSION

In this paper, we performed a quantitative examination of modern deep learning-based sound classification models and architectures as well as a lab exploration of a novel smartwatch sound awareness app with eight DHH participants. We found our best classification model performed similar to the state of the art for nonportable devices although requiring a substantially less memory ($\sim 1/3^{\text{rd}}$) and that the phone-based architectures outperformed the watch-centric designs in terms of CPU, memory, battery usage, and end-to-end latency. Qualitative findings from the user study contextualized our system experiment results and uncovered ideas, concerns, and design suggestions for future wearable sound awareness technology.

Acknowledgments

We thank Emma McDonnell and Ana Liu for their help. This work was supported by NSF Grant no: IIS-1763199. C

References

1. Bragg, D., Huynh, N., Ladner, R.E. A personalizable mobile sound detector app design for deaf and hard-of-hearing users. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility* (2016), ACM Press, New York, 3–13.
2. Braun, V., Clarke, V. Using thematic analysis in psychology. *Qual. Res. Psychol. 3*, 2 (2006), 77–101.
3. Findlater, L., Chinh, B., Jain, D., Froehlich, J., Kushalnagar, R., Lin, A.C. Deaf and hard-of-hearing individuals' preferences for wearable and mobile sound awareness technologies. In *SIGCHI Conference on Human Factors in Computing Systems (CHI)* (2019), ACM, Glasgow, UK, 1–13.
4. Foggia, P., Petkov, N., Saggese, A., Strisciuglio, N., Vento, M. Reliable detection of audio events in highly noisy environments. *Pattern Recognit. Lett. 65*, (2015), 22–28.
5. Goodman, S., Kirchner, S., Guttman, R., Jain, D., Froehlich, J., Findlater, L. Evaluating smartwatch-based sound feedback for deaf and hard-of-hearing users across contexts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2020), ACM, Honolulu, Hawaii, 1–13.
6. Gorman, B.M. VisAural: A wearable sound-localisation device for people with impaired hearing. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility* (2014), ACM, Rochester, NY, 337–338.
7. Guo, R., Yang, Y., Kuang, J., Bin, X., Jain, D., Goodman, S., Findlater, L., Froehlich, J. Holosound: Combining speech and sound identification for deaf or hard of hearing users on a head-mounted display. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility* (2020), ACM, 1–4.
8. Hershey, S., Chaudhuri, S., Ellis, D.P.W., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., et al. CNN architectures for large-scale audio classification. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2017), IEEE, New Orleans, LA, 131–135.
9. Jain, D., Findlater, L., Volger, C., Zotkin, D., Duraiswami, R., Froehlich, J. Head-mounted display visualizations to support sound awareness for the deaf and hard of hearing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015), ACM CHI, Seoul, Korea, 241–250.
10. Jain, D., Lin, A.C., Amalachandran, M., Zeng, A., Guttman, R., Findlater, L., Froehlich, J. Exploring sound awareness in the home for people who are deaf or hard of hearing. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), ACM, Glasgow, UK, 94:1–94:13.
11. Jain, D., Mack, K., Amrous, A., Wright, M., Goodman, S., Findlater, L., Froehlich, J.E. HomeSound: An iterative field deployment of an in-home sound awareness system for deaf or hard of hearing users. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20 (New York, NY, USA, 2020), Association for Computing Machinery, Honolulu, Hawaii, 1–12.
12. Jain, D., Ngo, H., Patel, P., Goodman, S., Findlater, L., Froehlich, J. SoundWatch: Exploring smartwatch-based deep learning approaches to support sound awareness for deaf and hard of hearing users. In *ACM SIGACCESS Conference on Computers and Accessibility* (2020), ACM, 1–13.
13. Kaneko, Y., Chung, I., Suzuki, K. Light-emitting device for supporting auditory awareness of hearing-impaired people during group conversations. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference* (2013), IEEE, Manchester, UK, 3567–3572.
14. Lu, L., Zhang, H.-J., Jiang, H. Content analysis for audio classification and segmentation. *IEEE Trans. Speech and Audio Process. 10*, 7 (2002), 504–516.
15. Matthews, T., Fong, J., Ho-Ching, F.W.-L., Mankoff, J. Evaluating non-speech sound visualizations for the deaf. *Behav. Inf. Technol. 25*, 4 (2006), 333–351.
16. Mazumdar, A., Haynes, B., Balazinska, M., Ceze, L., Cheung, A., Oskin, M. Perceptual compression for video storage and processing systems. In *Proceedings of the ACM Symposium on Cloud Computing* (2019), ACM, Santa Cruz, CA, 179–192.
17. Mielke, M., Brück, R. A pilot study about the smartwatch as assistive device for deaf people. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility* (2015), ACM, Lisbon, Portugal, 301–302.
18. Saunders, J. Real-time discrimination of broadcast speech/music. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings* (1996), Vol. 2, IEEE, Atlanta, GA, 993–996.
19. Shahzad, K., Oelmann, B. A comparative study of in-sensor processing vs. raw data transmission using ZigBee, BLE and Wi-Fi for data intensive monitoring applications. In *2014 11th International Symposium on Wireless Communications Systems (ISWCS)* (2014), IEEE, Barcelona, Spain, 519–524.
20. Sicong, L., Zimu, Z., Junzhao, D., Longfei, S., Han, J., Wang, X. UbiEar: Bringing location-independent sound awareness to the hard-of-hearing people with Smartphones. *Proc. ACM on Interact. Mob. Wearable and Ubiquitous Technol. 1*, 2 (2017), 17.
21. Yeung, E., Boothroyd, A., Redmond, C. A wearable multichannel tactile display of voice fundamental frequency. *Ear Hear. 9*, 6 (1988), 342–350.
22. Yuan, H., Reed, C.M., Durlach, N.I. Tactual display of consonant voicing as a supplement to lipreading. *J. Acoust. Soc. Am. 118*, 2 (2005), 1003.

Dhruv Jain, Hung Ngo, Pratyush Patel, Khoa Nguyen, and Jon Froehlich ({djain, hvn297, patelp1, akhoa99, jfroehli}@uw.edu), Computer Science and Engineering, University of Washington, Seattle, WA, USA.

Steven Goodman, Rachel Grossman-Kahn, and Leah Findlater ({smgoodmn, rachelgk, leahkf}@uw.edu), Human Centered Design and Engineering, University of Washington, Seattle, WA, USA.

[CONTINUED FROM P. 112] then, Hans had a list of the fastest computers and I had a benchmark that rated those computers; Hans approached me about putting it together and calling it the TOP500.

You have been designing software packages over the last 50 years. Why?

The software packages that I have been involved in designing have found their way into the fabric of how problems in computational science are solved. As computer architectures evolve over time, from scaler to vector to multicore to distributed memory to hybrid architectures, the software packages are among the first to adapt to the changes. They must be rewritten to embrace the architecture.

You can see this in the evolution and development of the packages. EISPACK was designed for scalar computers and LINPACK for vector architectures; LAPACK and the BLAS for use on cache-based and shared memory computers; ScaLAPACK and MPI were intended for distributed memory architectures, and PLASMA and MAGMA developed as the need for multicore and hardware accelerators (GPUs) entered the computer landscape. And today we are working on SLATE, which addresses the challenges of exascale-based computing. Along the way, there is the need for performance evaluation, and that is where the benchmarks fit into the picture.

The high performance conjugate gradients (HPCG) benchmark solves matrix problems using an iterative approach that manipulates sparse matrices.

Over the years, TOP500 rankings have evolved to include not just the LINPACK benchmark, but other ways to test how well computers can handle the sorts of tasks people need them to do.

The LINPACK benchmark has been in continuous use since the 1970s. It was born out of necessity, because it could quickly test the performance of vector subroutines, which served as a good approximation of performance for the rest of the LINPACK library. Thanks to the nature of the implementation, the LINPACK benchmark also served as a first-order approximation of other codes. That is partially due to the well-balanced hardware of the time, which offered plentiful bandwidth for every floating-point operation. Over the years, Moore's Law eroded the compute-to-bandwidth balance, resulting in a memory wall.

To reassess application needs in this new and different hardware regime, it is worthwhile to look at computational simulations. Many computational simulations involve heat diffusion, electromagnetics, and fluid dynamics. Unlike LINPACK, which tests raw floating-point performance, these real-world applications rely on partial differential equations (PDEs) that govern the continuous representations of physical quantities like particle speed, momentum, etc. These PDEs involve sparse (not dense) matrices that represent the 3D embedding of the discretization mesh. While the size of the sparse data fills the available memory to accommodate the simulation models of interest, most of the optimization techniques that help achieve close to peak performance in dense matrix calculations are only marginally beneficial in sparse matrix computations originating from PDEs.

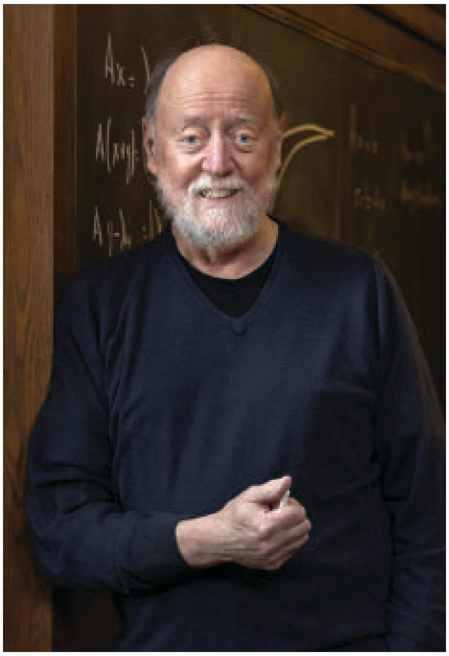
The authors of LINPACK—from left, Jack Dongarra, G.W. "Pete" Stewart, Jim Bunch, and Cleve Molar—photographed around Dongarra's car in 1978 in Downer's Grove, IL, near Argonne National Laboratory. Photo used with permission from Cleve Molar.

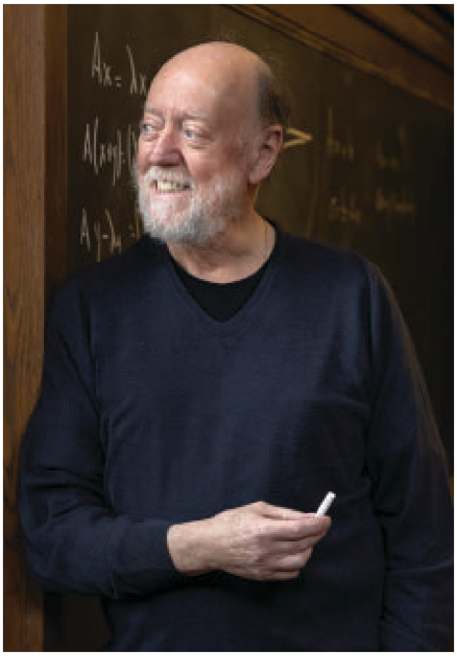
Why is that?

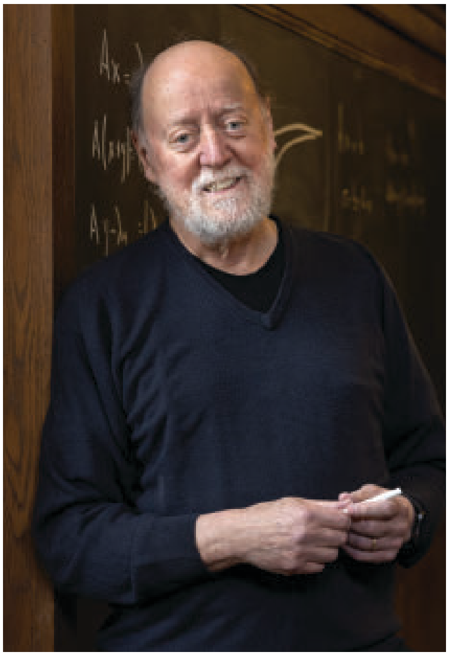
The TOP500 LINPACK benchmark can be characterized as a dense matrix doing dense operations. Machines that do floating-point operations efficiently are going to look good on this benchmark, even though most real-world problems don't actually require it.

So we developed a benchmark called the high-performance conjugate gradients, or HPCG. HPCG solves matrix problems not with a direct approach—not based on matrix multiplication where, if you have two matrixes of roughly order n, the number of operations is n^3 but the amount of data moved is only n^2. Instead, HPCG uses an iterative approach that manipulates sparse matrices, which shows the characteristics of the hardware better in terms of real applications.

Just to put that into perspective, if I were to run the TOP500 benchmark on a computer, I would expect to see performance reach 75% of the theoretical peak. The HPCG benchmark shows performance of 3% of the theoretical peak. That's our "dirty little secret": most applications are far from having reached







the theoretical peak of these high-performance computers. Still, it is a good way to expose the performance issues and look at how we might resolve and improve the situation.

Let's talk about what's new in the high-performance computing space. By the time we go to print, so-called exascale machines—which are capable of executing 10^{18}, or a billion-billion, floating-point calculations per second—may finally be on top of the latest TOP500 list.

Oak Ridge National Laboratory is installing a computer called Frontier, which will be the first exascale machine in the U.S. They're running the benchmark on it now, and we expect that will be completed by the time the next TOP500 list comes out.

What about the economics of high-performance computers?

The exascale machines are very expensive. I think the price tag for the machine at Oak Ridge is between $500 million and $600 million, and the Department of Energy (DOE) is engaged in a program that's developing three of these exascale machines—in addition to their investment in developing the applications and software that will run on these systems. The total price tag for that program over seven years is going to be around $3.6 billion.

These supercomputers are powerful, sophisticated scientific instruments, like the James Webb telescope. They enable simulation and provide the opportunity to push back the frontiers of science. Today, we use numerical computations to understand and predict the behavior of scientifically or technologically important phenomena—and therefore accelerate the pace of innovation.

It is important to note that every time computing power increases by large factors, new benefits open before us. The benefits of exascale computing—which range from creating novel, more efficient combustion engines and new energy solutions to advances in healthcare, biology, and storm prediction—could potentially impact every person. The benefits of exascale computing will flow from classical simulations but also from large-scale data analysis, deep machine learning, and often the integration of the three approaches.

"It is important to note that every time computing power increases by large factors, new benefits open before us."

Your work has been fundamental to a huge number of applications, from cloud computing to large-scale physics experiments. Are there things you've found surprising or exciting, beyond the work itself?

There's always something new to learn and use in solving the current problems. I have been fortunate to work with an incredibly talented international community of people over the years to develop algorithms, software, and standards that have helped shape the computational science area. That work could not have happened without those people.

Having students is instrumental in that respect, because it helps us push forward and explore multiple fronts. We try to do research, not just development, meaning we need to experiment and we need to fail, because that's an important part of the learning process. Sometimes, our students come to me in search of a research problem to work on, and when I give them one, they immediately tell me that they don't know how to do it. And I say, "That's perfect. That's exactly the point. I wouldn't give you a problem if you already knew how to do it." That's where the excitement is, learning new things and overcoming obstacles.

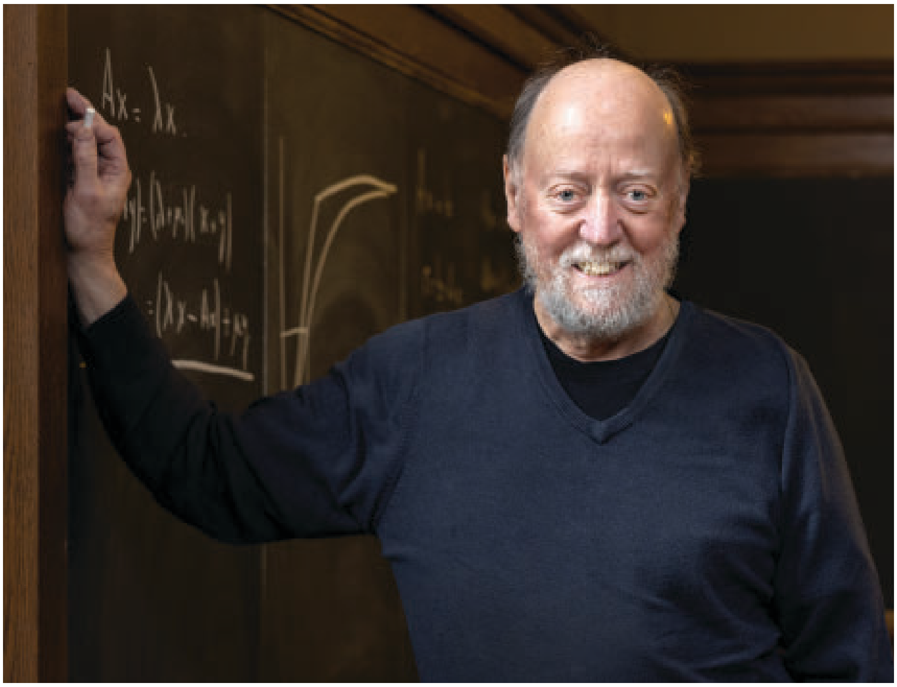
Leah Hoffmann is a technology writer based in Piermont, NY, USA.

DOI:10.1145/3530690

Leah Hoffmann

Q&A
Learning New Things and Avoiding Obstacles

ACM A.M. TURING AWARD recipient Jack Dongarra never intended to work with computers. Initially, the Distinguished Professor at the University of Tennessee and founder of the Innovative Computing Laboratory (ICL) thought he would be a high school science teacher. A chance internship at the Argonne National Laboratory kindled a lifelong interest in numerical methods and software—and, in particular, in linear algebra, which powered the development of Dongarra's groundbreaking techniques for optimizing operations on increasingly complex computer architectures.



"The software packages that I have been involved in designing have found their way into the fabric of how problems in computational science are solved," observes 2021 ACM A.M. Turing Award recipient Jack Dongarra.

Your career in computing began serendipitously, with a semester-long internship at Argonne National Laboratory.

As an undergraduate, I worked on EISPACK, a software package designed to solve eigenvalue problems. My role was helping to develop test problems and making sure things were working correctly. It was a wonderful environment. Then, I completed my master's degree, and they offered me a job.

"With my colleagues in Germany, Hans Meuer and Eric Strohmaier, we put together the first Top500 in 1993."

You began working on LINPACK around that time.

LINPACK was a National Science Foundation-funded project in the late 1970s that involved researchers at Argonne, University of New Mexico, University of California San Diego, and the University of Maryland. The goal was to design a software package for solving systems of linear equations that was based on state-of-the-art algorithms—and was portable, reliable, enhanced productivity, and provided efficient solutions to the scientific computer architectures in use at that time. As a way to measure efficiency, I constructed a benchmark to measure the performance of a computer when running the LINPACK software, which became the LINPACK benchmark. It appeared in a table in the LINPACK Users' Guide.

The table was the genesis of the TOP500 list of the most powerful computers.

With my colleagues in Germany, Hans Meuer and Eric Strohmaier, we put together the first TOP500 in 1993. Before [CONTINUED ON P. 110]

PHOTOS BY ALEXANDER BERG